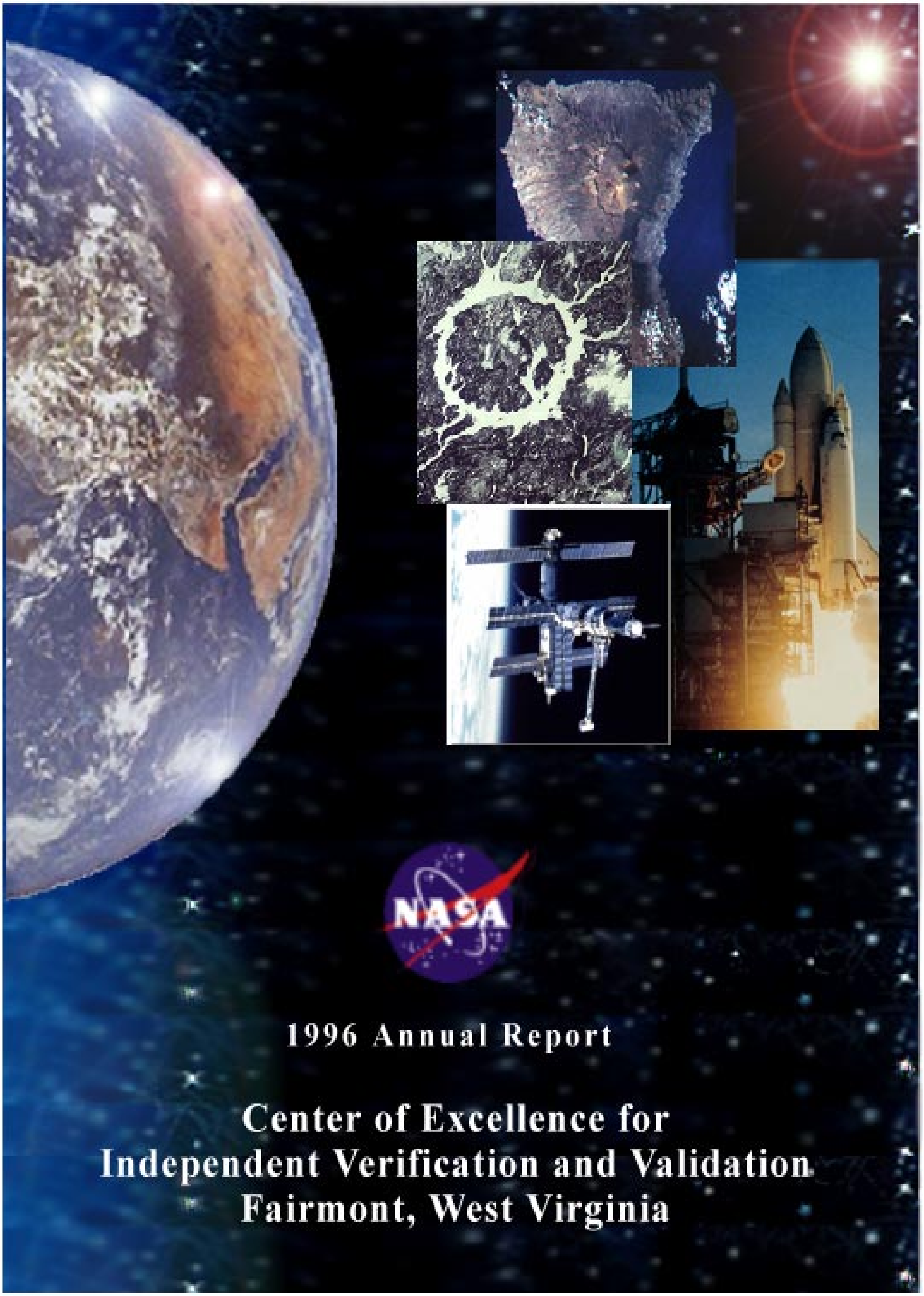


1996 Annual Report

Center of Excellence for Independent Verification and Validation Fairmont, West Virginia

NASA SS&T ANNUAL REPORT

Software Systems & Technology Office

DECEMBER 31, 1996

NASA/AMES Research Center
Software Systems & Technology Office
Fairmont, West Virginia 26554

Preface

This Annual Report is the first such report to be published highlighting significant achievements of NASA/Ames Research Center's Software Systems & Technology Office (SS&T) during the reporting year. More detailed information and/or technology interest may be obtained by contacting the designated Point of Contact for a specific area or visiting our Website at: [http://www.ivv.nasa.gov].

The SS&T was established in October 1993, to meet the need for an Agency Software Independent Verification & Validation (IV&V) capability. In October 1995, the management of the Facility, including the civil service and contractor support personnel, was transferred to NASA Ames Research Center (ARC) from NASA Headquarters, Office of Safety and Mission Assurance (OSMA), as part of the Ames' Center of Excellence for Information Technology (COE/IT). SS&T's **mission, roles, and responsibilities** are focused on **providing software IV&V for the Agency's major programs, as requested by the OSMA, and maintaining an Area of Excellence in software IV&V to enhance and strengthen national competitiveness**.

The SS&T has four major activities:

- **Software IV&V Analysis and Assessment:** Work in this area is focused on ensuring that the flight safety and mission success for NASA's major programs are not compromised by software errors. Significant accomplishments over the past year have been in the strengthening of SS&T's IV&V capabilities relative to personnel, both civil service and contractor-support, development of generic testbeds representative of the project's operational environment for early evaluation of software and establishment of metrics, and development of the process leading to software reuse. Efforts have been initiated to provide critical systems software analysis capabilities and software assessments to project managers requesting the service. Funding for this effort is largely provided by OSMA.

- **Basic and Applied Research and Development:** A software engineering research program has been initiated to investigate and pursue advanced software methods and processes which will result in automated software IV&V throughout the development process. This capability will provide an end-to-end software engineering approach which will significantly increase the software reliability of the end product, minimize software rework, and increase the potential for the reuse of validated software modules accessible through object-oriented, relational databases. The initial effort is being sponsored by COE/IT.

- **Technology Transfer:** More emphasis is being placed in this area over the next calendar year to ensure that any technologies developed under contracts sponsored by NASA are identified and transferred to industry in a timely manner. Working relationships have been established with the West Virginia High Technology Consortium Foundation's (WVHTC Foundaition) Institute for Software Improvement to identify and develop a long-term program with significant annual milestone achievements. The detailed plans for this cooperative relationship are expected to be completed by early April 1997. In addition, a process has been initiated to identify the technologies of benefit to potential user organizations located in West Virginia and strategically match those requirements against the technologies developed across the Agency. It is expected that this process will be completed by early August 1997.

- **Educational Outreach and Training:** Significant achievements have been accomplished in this area working with the West Virginia's educational community and the WVHTC Foundation. NASA educational resources will be available at the Facility no later than March 1997, as an adjunct to the NASA Software Technical Library (NSTL). NSTL will be a Satellite Library to the Ames/Moffett Main Library and will provide immediate access to the technical documentation required to conduct the Agency's IV&V work as well as its supporting functions. West Virginia University (WVU) will operate NSTL under contract to ARC. Sharing of technical journals, periodicals, and related technical documentation between ARC and WVU will reduce cost and duplication of effort to both organizations. In addition, training agreements have been established between WVU, Fairmont State College, and ARC to provide pertinent software assurance training to Agency personnel and to upgrade and enhance the skills of SS&T personnel in their area of expertise. Initial funding for software assurance training is being provided by OSMA with ARC providing the start-up funding for NSTL. Educational resources have been coordinated and provided by NASA's Office of Public Affairs (Educational Programs and External Affairs).

Emphasis during the next year will be placed on strengthening the infrastructure required to be a nationally recognized Center of Excellence in IV&V through an integrated, focused program. For further information on how the SS&T functions, refer to the NASA COE - IVV Business Plan.
Metrics have been established to measure SS&T's progress:

- Increase the number of NASA programs/projects that seek out SS&T's IV&V capabilities to maximize system software safety in a cost-effective manner.
- Increase the number of nationally and internationally recognized personnel who seek employment at the SS&T.
- Gain commitment from at least one new major program/project to have IV&V work performed at the SS&T through the integrated efforts of NASA, WVU, and the WVHTC Foundation.
- Gain recognition from industry that NASA can provide IV&V service in a timely, cost effective manner and commit them to entering a joint collaborative agreement with NASA for such service.

Dr. Henry Lum, Ames/Fairmont Facility Director
(henry.lum@ivv.nasa.gov)

Editor's Note: The *Related Publications* section of each article is designed to reflect publications produced by that project/activity. The reference numbers refer to the location of the publication within Appendix B.

TABLE OF CONTENTS

The Software Systems & Technology Office Independent Verification and Validation Program

International Space Station Independent Verification & Validation

Earth Observing System Data and Information System

Validation of Software Metrics for Independent Verification and Validation

Verification and Validation of the Information Sharing Protocol

The Software Systems & Technology Office Independent Verification and Validation Program

Objective

To ensure the delivery of a software product that:

- Remains within budget
- Stays within schedule
- Meets all requirements

Approach

Independent Verification & Validation (IV&V) is the application of a variety of techniques, often supported with automated software tools, to evaluate critical/complex software. IV&V consists of two distinct but complementary processes.

- Verification is the process of determining whether or not the products of a given phase of the software development cycle fulfill the requirements established during the previous phase.
- Validation is a process of evaluating software at the end of its development process to ensure compliance with software requirements. This process ensures that the software produces expected system behavior when subjected to anticipated events and does not produce unexpected system behavior when subjected to unanticipated events.

These techniques are applied by an independent organization or contractor. The "independent" term in IV&V, as implemented by the Software Systems & Technology Facility (SS&T), is defined as having three explicit attributes:

- Technical: Technical independence requires that the IV&V team utilize personnel who are not involved in the development of the software and system.
- Managerial: Managerial independence requires that the IV&V responsibility be vested in an organization outside the contractor and program organizations that develop the software systems. Managerial independence also requires that the IV&V team independently decides:
  1. Areas of the system to analyze and test
  2. Techniques to be used in the IV&V
  3. Schedule of activities to be performed (within the framework of the system schedules)
  4. Technical issues to be acted upon
- Financial: Financial independence requires that control of the IV&V budget be vested in an organization outside the contractor and program teams that develop the system. Currently, financial independence is not possible in the National Aeronautics and Space Administration (NASA) accounting structure. Software IV&V is financed by the programs undergoing IV&V.

Significance

Software systems continue to grow in size and complexity. As a result, the cost of developing software has been increasing and in some cases has surpassed the cost of developing hardware. In response to these increases, management has become increasingly concerned about the feasibility of developing software within initial cost estimates and on schedule, while achieving and maintaining high quality. This concern has proved justified time and again in programs across all agencies in the government as well as in private industry.

Two IV&V programs were housed in the facility over the last two years: International Space Station (ISS) and the Earth Orbiting System Data and Information System (EOSDIS).

Related Publications

14

Point(s) of Contact

John Hinkle
(john.hinkle@ivv.nasa.gov)

International Space Station Independent Verification & Validation

Objective
To perform IV&V activities for catastrophic/critical/ high risk software systems.

Approach
Given the huge software integration challenge in the ISS, IV&V has concentrated much effort to ensure processes are valid and in place, ensuring architectural concepts are implemented across systems, and in vertical (e.g., Electrical Power) software verification & validation (V&V).
Normal IV&V interaction with the program is through various ISS teams, on an informal basis, for conveying IV&V analyses results in a timely manner to the program. When ISS IV&V recommendations are not accepted or implemented in a timely fashion, IV&V conveys recommendations to the ISS program through an established formal path.
The staged development has resulted in IV&V functions being performed simultaneously at system requirements, software requirements, software preliminary design, and software detailed design phases for different software. Finally, the Integrated Product Team/Analysis & Integration Team structure has caused IV&V to develop an approach that is more informal and more responsive to the success of providing timely suggestions to the program.

Significance
The ISS involves the United States (US) and four international partners assembling a station in low earth orbit during 44 flights taking place over 4.5 years, beginning in November 1997. Since the ISS is developed and assembled in stages, the software required for different stages undergoes phased development. At times, there are software components spread out across the entire software developmental life-cycle requiring potential ISS IV&V involvement. The resulting software challenges of integrating components developed across the US and world are virtually unprecedented. In answer to those challenges, IV&V has involved itself in every area of flight software development.

Accomplishments
The IV&V team measures its accomplishments by the implementation of its suggestions. Through formal review comments, white papers, issues sheets, and presentations, IV&V has affected numerous, significant, positive changes in the ISS program. The IV&V team has:

- Focused program attention on the lack of integrated software schedules resulting in the implementation of integrated software schedule tracking
- Increased radiation tolerance in the program's main computer by raising issues that resulted in a chip alteration
- Focused program attention on the risk of using Matrix-X to replace most software testing at two of the principal developers
- Removed an unnecessary simulation capability from the onboard guidance software, relieving the program from the expense of development, and eliminating the risk of non-operational code
- Caused the program to change its certification approach for simulations used in the acceptance of flight software
- Raised the issue at the Russian software review that their Software Requirements Documents lacked necessary content, the Russian Space Agency was instructed to deliver new documents
- Identified significant developmental risk assumed by not completing and baselining the software safety requirements, two tiger teams were created to respond to this issue
- Assessed the initial stages' command and control software to be in a Red Status and recommended mitigation approaches - resulted in replanning the software
- Established that the program was significantly behind in the identification of hazardous commands which would result in costly rework of software late in the development life-cycle, a tiger team was sent to rectify the situation
- Identified significant risk in the development of the Portable Computer System (PCS), resulted in the program completely changing the PCS management structure and responsibilities
- Caused the program to implement a certification approach to the Commercial of the Shelf (COTS) Run Time Environment

The benefits of IV&V on a program are not readily measured by established metrics. While all evidence points to IV&V as a significant contributor to the safety and success of programs, the benefits are, for the most part intangible. The ISS IPT structure contributes to this problem because most software products are reviewed in the informal, team structure. Although informal contributions are the hardest to report and quantify, they effectuate the most significant impact for little cost. Through informal contribution, the IV&V team has:

- Streamlined SS&T problem reporting mechanism to reduce management overhead and follow a proven process from the SS program
- Identified 400 disconnects between two Interface Control Documents
- Defined the stage and software formal review success criteria now specified in the Prime Developer's Software Development Plan
- Examined, through modeling, an algorithm in the Electrical Power System software requirements, provided requirements clarification to the developer's programmers
- Briefed internal ISS assessment teams examining the software on the current status of the program software
- Caused Government Furnished Equipment (GFE) software developers to be required to present same data (schedule, status, metric) at Software Monthly Reviews as the contract developers, increased visibility into the GFE status

Future Plans

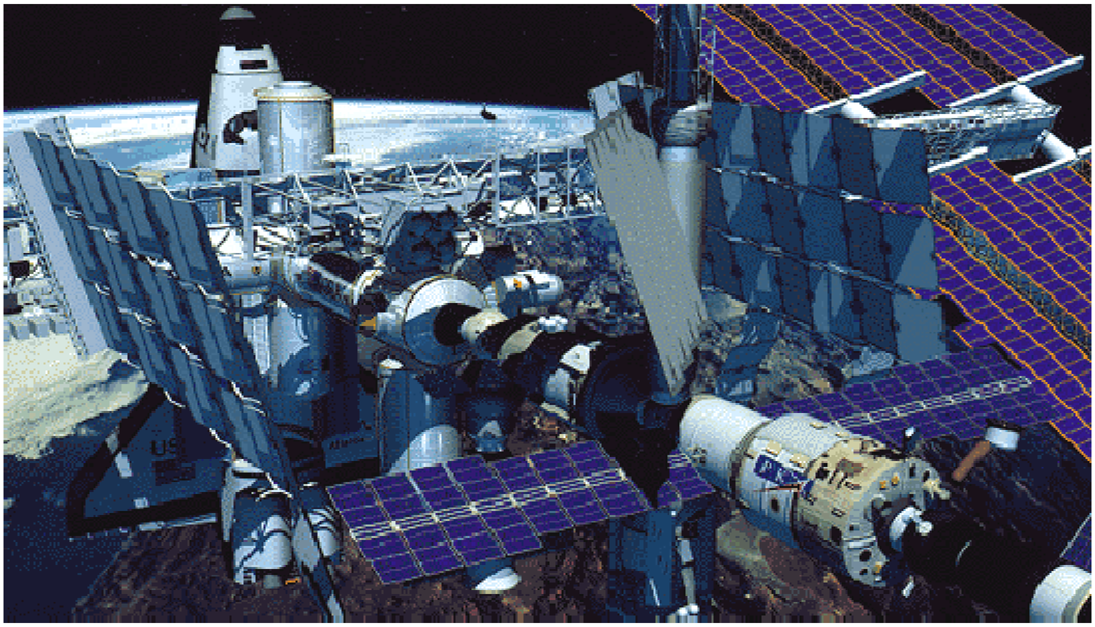
Contracts are in place to ensure that IV&V will be performed throughout the development of the ISS until assembly is complete in 2003**.**

Related Publications

58

Point(s) of Contact

John Hinkle
(john.hinkle@ivv.nasa.gov)



Initial ISS activities on Independent Verification and Validation of software appear to be following a logical and reasonable approach. The IV&V contractor seems to be well on board and establishing relationships with the program so that they can have access as the work proceeds. They have decided not to attempt to bite off more than they can chew and have developed what appears to be an acceptable approach to the job. Having half their work force at the Johnson Space Center is good and is vital to their effectiveness. Their approach of bringing up issues at the lowest reasonable level and escalating up the chain of command as necessary is well advised and should be effective.

The initial IV&V work focused on a number of programmatic issues and provided good insights into some real program problems. Once requirements are finalized, it is hoped that IV&V efforts will turn to analyses of the software itself. NASA should build upon the good start that has been made in the ISS IV&V effort.

The Aerospace Safety Advisory Panel, in their last report on the status of the ISS program.

Earth Observing System Data and Information System

Objective

To develop analytical tools and techniques for the integration and certification of critical software.

Approach

Since April 1994, a portion of the personnel responsible for conducting software IV&V for EOSDIS has been located at SS&T. The EOSDIS IV&V contract is managed by Goddard Space Flight Center (GSFC) personnel.

Significance

As a result of Project Issue Tracking System (PITS) deployment, several benefits have already been realized. PITS now serves as the central issue repository which hosts major IV&V findings and associated information. This repository serves as the foundation and source of information when preparing for IV&V findings meetings and in generating reveiw item discrepancies following milestone reviews. The insertion of this tool technology has not only improved the IV&V process, but due to inherent management indicators and workflow mechanisms, drives it. In addition to the PITS client/server application, accessibility of PITS repository data is now available via the EOSDIS IV&V Homepage. This is expected to provide several benefits:

- Wide Area Network (WAN) access to requirements analysis and test information for geographically dispersed users (Fairmont, Greenbelt, and GSFC)
- Variance analysis of requirements analysis data and test results (following Discrepancy Report closures)
- Automatic submission of Structured Query Language (SQL) queries against databases to find needed information
- Dumping data for inclusion in document deliverables or placement on homepages
- Automated accumulation of metrics by an SQL query
- Test Buddy use for recording results for each test conducted at remote test sites

Accomplishments

The ISS IV&V effort has gained benefits from the on-site activities of EOSDIS IV&V personnel. Tools and techniques from EOSDIS have often found their way into use by ISS IV&V. A Memorandum of Understanding exists between the contractor and West Virginia University (WVU) that has furthered SS&T's research goals.

In late December 1995, Task 4B personnel initiated a design of the PITS database schema and data entry interfaces. On January 11,1996, the initial data entry prototype was released for feedback and informal training was also provided to the IV&V analysts. Since that time, several PITS client/server tool releases have been made and current functionality includes the following:

- Ad-hoc query capabilities allowing searches on metadata items, dates, and all text strings
- Issue reporting including metrics and aging reports
- Workflow messaging to streamline processing and statusing of existing issues
- Review Item Discrepancy generation, modification, and mailing
- Monthly database snapshots to support trend analysis

Incremental releases of the client/server Automated Requirements Database (ARDB) and the Test Management Database (TMDB) were installed for use. These tools allow Personal Computer (PC) Windows clients to access database data over the WAN that is maintained in the Requirements Traceability Management/Oracle database and the Integrated Support Environment Sybase SQL Server database. The ARDB is being integrated into the Requirements Analysis Task to improve the existing process. Likewise, the TMDB is being incorporated into component, integration and test, and system certification test activities to enhance information accessibility and dissemination. A PC has been configured at GSFC with both the ARDB and TMDB applications so that GSFC personnel have real-time access to both requirements analysis and test information.

Point(s) of Contacts

Randy Hefner
(randy.hefner@ivv.nasa.gov)

John Hinkle
(john.hinkle@ivv.nasa.gov)

Validation of Software Metrics for Independent Verification and Validation

Objective

To better understand software and the software development process by studying software measurement.

To examine metrics for meaningfulness in terms of the scale assignable to the metric by the rules of measurement theory and the software dimension being measured.

Approach

In software measurement, it is known that the product/process that cannot be measured cannot be controlled. The products and processes of software development can only be evaluated if an appropriate set of metrics exist for measuring them. This set of metrics must be validated, to ensure that the metrics do indeed measure those properties of the product or process that they purport to measure.

The Research Team approach is to develop a framework for validation of software measures, to provide a consistent viewpoint for the software properties of interest. Existing measures proposed in the literature can then be evaluated according to this framework. The categorization of the measurements by some meaningful taxonomy is the first step toward a better understanding of software measurement.

To date the research has concentrated on measurement of Object Oriented (OO) software. The characteristics of OO software offers a sensible starting point for such a categorization of measures, as the characteristics (i.e. 'dimensions') of OO software have been explored in detail in the literature. A meaningful taxonomy of these dimensions would capture the basic nature of the OO space, and provide a foundation for the validation of OO metrics.

Significance

Software measurement is of particular importance to IV&V. Timely feedback from the IV&V effort into the development process is particularly crucial for the effectiveness of IV&V, and the credibility of the IV&V agent. Feedback is needed both on actual and potential problems. An important mechanism for identifying problems is the measurement of leading indicators. Finding a valid set of leading indicators is therefore an important task for the success of IV&V. The dangers of randomly selecting variables to include in statistical processes to show causality are well known. In the software engineering community, metrics are accepted as predictors with little or no theoretical validation [1].

Accomplishments

A taxonomy has been developed based upon characteristics of OO software gathered from literature. This taxonomy allows easy viewing of the gaps and redundancies in the OO measures. It also clearly differentiates among taxa so that there is no ambiguity as to which taxon a measure belongs.

The taxonomy has been populated with 32 metrics that have been validated using measurement theory with Zuse's augmentation.

Future Plans

Research personnel will continue to participate in the Joint Logistics Commanders Joint Group on Systems Engineering efforts to define software product measures. As opportunities arise, the research team will expand their work to additional industry (contractor and academia) and government working groups.

References

1. Fenton, Norman E, *Software Metrics A Rigorous Approach*, Chapman & Hill, London, 1991

Related Publications

20, 70-73

Point(s) of Contact

Ralph D. Neal
(ralph.neal@ivv.nasa.gov)

Verification and Validation of the Information Sharing Protocol

Objective

To develop expertise in validating and verifying distributed client/server software and to automate the V&V process to significantly reduce or eliminate the current manual V&V efforts.

Approach

Information Sharing Protocol (ISP) is the telemetry acquisition and data sharing system being deployed in the new Mission Control Center (MCC). Both SS and ISS flight support software planned for the new control center is based on ISP. The ISP client/server architecture enables distributed flight support software to share spacecraft telemetry and higher order (ground generated) information. It is also the mechanism being employed to distribute mission information outside the control center. The research will involve three phases of activity:

1. Assessment of the automatic fault tolerance of ISP through the application of manual V&V testing procedures
2. Implementation of solutions to robustness problems identified in Phase 1
3. Investigation and application of automated approaches to V&V testing of distributed client/server flight support software

Significance

The current MCC software certification processes involve many hours of manual, often repetitive, execution of software test plans by MCC flight controllers. These test plans must be repeated for recertification of the software anytime the software is recompiled whether due to a change in the software or in the MCC platform. Automated software testing tools and the new processes that define how and when to use them would significantly reduce the manual effort currently involved in the certification/recertification process. The tools and processes would also improve the quality of the software by providing a more complete test of the software's inputs and data paths than is possible with manual execution of test plans.

Accomplishments

- Completed the identification of the failure scenarios that affected the ISP distributed client/server environment
- Provided development and testing support to Loral for the ISP heartbeat software to help address these failure scenarios
- Allowed the MCC to support the Space Transportation System (STS)-76 mission from the new MCC for entry operations and the STS-77 mission for the entire mission
- Identified commercially available automated testing tools
- Completed some of the tool evaluations, Mercury Interactive has not been responsive
- Evaluated the testing procedures for background computations for the Instrumentation and Communications Discipline Oimon application
- Provided recommendations for designing testability into the applications to facilitate off-line testing of the core functionality
- Designed the ISS Calibration application aggressively factoring in off-line V&V of the core functionality to apply the lessons learned from Oimon review and recommendations
- Designed the Attitude Determination Control disciplines ISS Hilo application based upon the conclusions from both the Oimon and ISS Calibration comp testing procedures
- Included cases, in the Hilo regression testing, that exercised the core logic independently of ISP
- Redesigned the ISS calibration comp to include specific "state" and "processing" modules, enabling ISP-independent regression testing

Point(s) of Contact

Siamak Yassini
(siamak.yassini@ivv.nasa.gov)

Assessment of Critical
Systems Software for
NASA Strategic Enterprises

Assessment of the Cassini
Command and Data
Subsystem

Critical Sequence
Rollback Analysis

Independent Verification &
Validation of the Cassini
High Speed Simulator

Standardized Method for
Validating Compilers
Used by Cassini

Analysis of the Software
Development Environment
and Processes in the Flight
Software for the Cassini
Mission

Assessment of Critical Systems Software for NASA Strategic Enterprises

Objective

To perform technical independent assessments of NASA software products and processes.
To provide assurance that safe and reliable software is being provided to the Strategic Enterprises.
To help program management achieve successful software development.

Approach

Four software assessments are performed:

1. Systems software enhancement assessments evaluate basic requirements, design, V&V, IV&V, and operations of new enhanced systems
2. Software life cycle development assessments identify risks associated with mission safety during all software development life cycle phases and make recommendations for corrective action
3. Mission Software Readiness Assessments (MSRA) in preparation for Flight Readiness Review (FRR) and as input for the Office of Safety Mission Assurance (OSMA) signature of the Certificate of Flight Readiness (COFR) independently assesses critical software changes and anomalies
4. Criticality Analysis and Risk Assessments (CARA) identify software components that are more critical with higher developmental risks and makes recommendation for IV&V activities

Significance

Leading-edge capabilities in fields such as software assessments, IV&V, and software development methodologies/tools are the foundation of a successful, highly-complex technical program.

Accomplishments

For Human Exploration and Development of Space:

- Reviewed software requirements, life-cycle processes and products, and evaluated the V&V processes including integration testing for SS Avionics, and Space Shuttle Main Engine (SSME) Controller software
- Analyzed the critical Software CRs and DRs with Severity 1 and 1N only per each flight software Operational Increments (OI)
- Performed systems software assessments of two SS system enhancements: Global Positioning System (GPS) and Multi-Function Electronic Display System (MEDS)
- Evaluated critical software functions during the life cycle development and recommended corrective actions
- Performed analyses of the critical Software CRs and DRs with Severity 1 and 1N only flight software OI for OI-25
- Prepared MSRAs for STS 70-81 for the FRR as input for the OSMA signature of the COFR

For Space Science Missions:

- Performed independent assessments of critical software elements for two near term NASA Space Science Missions : Cassini and 1998 Mars surveyor program (Mars ‘98)
- Performed assessments of the critical software elements for the flight, ground, and operating system of the Cassini spacecraft
- Performed assessments of critical software elements for the Mars ‘98 Orbiter and Lander in the areas of flight, ground, and operating system software development processes

Future Plans

- Continue to perform technical independent assessments on high critical risks software components in support of SS Launches and future Space Science missions
- Provide assessments and CARA for the new launch Processing System at Kennedy Space Center in support of the SS launches
- Review the software processes and identify the software components that are more safety critical for IV&V activities
- Perform an independent assessment and review of the software life cycle development processes of the Small Satellite Technology Initiatives as requested by other agencies

Point(s) of Contact

Siamak Yassini
(siamak.yassini@ivv.nasa.gov)

Assessment of the Cassini Command and Data Subsystem

Objective

To determine if the use of theorem proving within a formal model can be used to generate test cases for testing the actual system being modeled.

Approach

The research team has determined the functioning of the scheduler and interrupt handler through analysis of the Cassini Command and Data Subsystem (CDS) source code. This information is being used to create a formal model of this part of the CDS software using the formal specification language, prototype Verification System (PVS), which is based on a classical, typed higher-order logic. The PVS system includes support tools and a theorem prover. The research team has previously used formal model checking to generate test cases for the actual system. The test data and the model checking have been used jointly to maintain the fidelity of the model with the actual system as it is being developed. This project will attempt to use theorem proving within the formal model in a similar fashion, to generate test cases for the CDS and to increase the fidelity of the model to the CDS, as illustrated in the figure.

Significance

Formal methods have been promoted as an effective means of developing software that requires a high degree of assurance. Yet to date, formal methods have not made a significant impact on software development practices. The difficulty of maintaining the fidelity of the formal model with the system as it undergoes changes in requirements and specifications during development has contributed to the failure to be included in the state of the practice. The union of formal models with test case generation will help to alleviate the difficulty of maintaining fidelity, and will provide useful information in the form of test cases to the development team.

Accomplishments

The team has analyzed source code from the Cassini CDS related to the scheduler and interrupt handler functions. The team identified that the millisecond interrupt was not functioning properly and corrections were made. The team has also developed a preliminary PVS model of the scheduler and interrupt handler functions.

Future Plans

The team will continue to build the formal model of the scheduler and interrupt handler functions of the Cassini CDS. As the model matures, the team will attempt to use theorem proving to produce test cases for the CDS. These tests will be executed during Cassini subsystem and integration testing. The resulting data will be used to improve the model and generate additional test cases.

Related Publications

5, 6

Point(s) of Contact

Edward A. Addy
(edward.addyivv.nasa.gov)

Critical Sequence Rollback Analysis

Objective

To determine if a dual redundant distributed computing system, utilizing a mark and rollback error recovery scheme, could be adequately and reliable tested.

Approach

Three questions were addressed in the completion of this task:

1. How have other schemes been validated?
2. How do these validation schemes compare to the ones presently in place?
3. What should the validation team do to improve their work?

Mark and rollback schemes are used to make software fault tolerant when the errors that need to be responded to are not known at the implementation of the program. The schemes function by rolling program execution back to an earlier location where no errors are in evidence. The code is then re-executed forward from that point. The advantage of this process is that the entire program sequence need not be re-executed. The project began with a literature search making use of the extensive knowledge base at the NASA Software Technical Library, the local environment, and the diverse material available on the Internet. During weekly meetings with the project, progress and problems were discussed to keep all work focused on the project goals. This feedback was useful and allowed changed and/ or new requirements to be readily incorporated into the study. Monthly Management Meetings provided the opportunity to report to more of the project personnel resulting in additional feedback on progress to date.

Significance

A specific validation modeling scheme that could be used to validate the Mark and Rollback application was identified. This scheme allows the implementation to be exhaustively tested to validate the requirements and design in cases where the state space is not too large. When the number of states is prohibitively large, two approaches can be used to provide increased state space coverage.

1. The state space can be judiciously partitioned into equivalence classes dividing the validation problem into several small problems that can be handled.
2. Powerful search techniques can be used provide additional assurance that the implementation is error free although it cannot guarantee it.

It is possible to validate high level requirements and design using formal theorem proving techniques. By showing that simulation modeling tools could be used to validate this system, the actual implementation can now be validated.

Accomplishments

The search turned to what specific options could be pursued to validate mark and rollback systems, having found no specific work detailing how other systems of this nature were validated. Accordingly, the mark and rollback problem, as it was framed for the system to be validated, was shown to behave as a communications system problem. First, the communications protocol involved was identified. Then several validation modeling schemes were identified that could be used to validate the actual system implementation.

Future Plans

In the future a modeling scheme will be developed to validate the mark and rollback process using one of the identified validation modeling tools. This work has applications in other areas of software development as well.

Point(s) of Contact

Dr. Francis Schneider
(francis.l.schneider@jpl.nasa.gov)

Independent Verification and Validation of the Cassini High Speed Simulator

Objective

To produce an operationally useable and user-reliant ground-based spacecraft simulator to minimize adverse risk from commands sent to the Saturn-bound Cassini spacecraft.

To test and check spacecraft commands and computer instructions that will be sent to the spacecraft for command and control purposes.

Approach

In order to integrate the IV&V agent into the development and testing processes of the High Speed Simulator (HSS), there must be a thorough and complete understanding of the development processes and activities. This will allow an understanding of the concerns and magnitude of the simulator's development problems.

Significance

Evidence[1] suggests that IV&V involvement in the development process can add a significant amount of measurable and value-added techniques to achieve:

- Higher product quality
- Better conformity to sustainability needs and standards
- Greater productivity from the remaining available schedule
- Finer degree of user-satisfaction
- Increased usability of the product

Concerns are raised and the development teams' responses to those concerns assures NASA of a superior product. If the development teams' responses to the concerns are not adequate, NASA has the opportunity to take appropriate actions to minimize those concerns. If an IV&V agent was not present at the development site, NASA could potentially be unaware of what is being developed until delivery. At which time, unexpected and unwanted product attributes would result in added costs to remedy or abandon a tool that has been a large investment.

Accomplishments

A requirements matrix was created that describes the detailed requirements that will be included in each HSS delivery. This matrix allows the test team to determine the thoroughness and completeness of each delivery.

Concerns have been communicated to development and user management teams about progress and development efforts with respect to short-term objectives and long-term needs. This allows the focus of resources on areas that might have otherwise been neglected. Three examples of where this task has had a direct impact are:

- Added several sections to the HSS Users Guide (examples of output files, storage requirements, and help information)
- Improved documentation for the HSS graphical user interface and ATPF_GEN
- Facilitated additional user testing on the HSS

Future Plans

There will be continued monitoring, assessing, and reporting of the development efforts of the HSS until launch of the Cassini spacecraft. This will provide additional assurance that the development effort will produce a quality product with the expected results.

References

1. Wallace & Fujii *Software Verification and Validation: Its Role in Computer Assurance and Its Relationship with Software Project Management Standards* (1989) (p 15-17) (NIST Special Publication 500-165)

Point(s) of Contact

Frank Balay
(frank.balay@jpl.nasa.gov)

Kathryn Kemp
(kathryn.kemp@ivv.nasa.gov)

Standardized Method for Validating Compilers Used by Cassini

Objective

To establish a standard way of validating and performing acceptance testing on new or updated compilers for the 1750A development platform.
To mitigate the risk that is introduced each time the Ada compiler is changed.

Approach

Testing Research - Investigate all commercial Ada compiler test suites to be used on the TLD Ada compiler. By using a COTS test suite for Ada compilers, a benchmark can be established for the existing version of the Ada compiler on the 1750A platform. The test suite that will be used for establishing the benchmark for the TLD Ada compiler is version 2.0.1 of Ada Compiler Validation Capability (ACVC) test suite. Once this benchmark is established, the test suite can be run whenever a change has been made to the compiler or operating system. The test results can then be compared to the original benchmark. The differences between the results of the benchmark run of the test suite and the results of the test suite when the compiler or operating system is changed will highlight the effect the change has in the application program.
Prototype - Investigate the use of a prototype as a tool to help integrate new and updated versions of the Ada compiler. In addition to the ACVC testing to establish a benchmark, a prototype will be developed for testing. This testing will be closer to the actual application because it will include the same type of functionality requirements as the application itself. This will be a more in depth level of testing than the ACVC test suite and will have the advantage of not having the application's hardware dependencies. This will be accomplished by disabling the hardware dependencies to modify the application, modifying existing prototypes, or a combination of both.

Significance

Ada Compiler Testing Research - Without establishing a benchmark with the test suite, Jet Propulsion Laboratory (JPL) would have to use the application itself for testing the Ada compiler. By establishing a benchmark test suite, the problems caused by a change in the Ada compiler could then be isolated.
Prototype Testing Research - This provides additional testing and an additional benchmark from the ACVC testing described above. This benchmark would provide depth that the ACVC test would not have. In doing so, the benchmark would be much closer to the needed functionality of the application itself without the hardware dependencies. Like the ACVC testing above, this benchmark would be frozen so that problems created by changing the Ada compiler could then be isolated.

Work in Progress

Ada Compiler Testing Research - Provisions to complete the ACVC testing and analysis are planned for the next year. When analysis is complete, a second trip will be necessary to complete the testing at JPL and a final report of the test results will be delivered. The automation and results of the test suite will then be the benchmark used to test any changes made to the Ada compiler.
Prototype Testing Research - When the necessary information is received, plans for the prototype testing will continue and a prototype benchmark will be developed.

Point(s) of Contact

Roy M. Kincaid
(mac.kincaid@ivv.nasa.gov)

Kathryn Kemp
(kathryn.kemp@ivv.nasa.gov)

Analysis of the Software Development Environment and Processes in the Flight Software for the Cassini Mission

Objective

To use the research skills and expertise of SS&T personnel to give practical advice to software developers in order to enhance the maintainability of the software used for the Cassini mission.

Approach

The flight software subsystems of Cassini will be analyzed using the following techniques:

- Interviews with the software development teams
- Observations for the software development teams in action
- Analyses of the current development environment with regard to the maintainability of the development environment for the duration for the Cassini mission

These activities will then be combined with current research ideas and practices to enable the Cassini mission software developers to take practical steps in enhancing the maintainability of the Cassini software.

Significance

The Cassini mission, like all NASA missions of long duration, faces two major problems:

1. The maintenance of a spacecraft system for the length of the mission and beyond.
2. The dynamics of spacecraft software responsibility and personnel for the lifetime of such a long mission.

The significance of this is the benefit that will be provided to the Cassini mission by enhancing the maintainability of the software systems and the science onboard the Cassini spacecraft for the mission lifetime. This will also benefit other NASA missions that face similar issues regarding longevity, maintenance, and personnel management.

Work in Progress

The main results will be coalesced into two reports that describe:

- How the development environment can be enhanced to ensure maintainability during the mission lifetime
- How the processes for developing the flight software can be enhanced to ensure maintainability during the mission lifetime

When the activity has been successfully completed it will be reviewed by the Cassini mission. The results of the activity will then be disseminated to other NASA missions and other industries facing similar problems maintaining computer systems and environments for a long periods of time.

Point(s) of Contact

Kathryn Kemp
(kathryn.kemp@ivv.nasa.gov)

.

Verification and Validation
Within Reuse-based Software
Engineering

Software Optimization and
Reuse Technology

Langley Research Center
Wind Tunnel Control
Systems Environment

Mission Operations Systems

Reusable Objects Software
Environment

Verification and Validation Within Reuse-based Software Engineering

Objective

To determine the usefulness and methods of performing V&V within reuse-based software engineering.

Approach

V&V is used to increase the level of assurance of critical software, particularly that of safety-critical and mission-critical software. V&V is a systems engineering discipline that evaluates the software in a systems context, and is currently applied during the development of a specific application system. In order to maximize the effectiveness of V&V within reuse-based software engineering, V&V must be incorporated within the domain engineering process. One model for reuse-based software engineering is the Software Technology for Adaptive Reliable Software (STARS) Two Life-Cycle Model. This model assumes a domain-specific, architecture-centered approach to software reuse, and includes the two life-cycles of Domain Engineering (DE) and Application Engineering.

Significance

Failure to incorporate V&V within DE will result in higher development and maintenance costs due to losing the opportunity to discover problems in early stages of development and having to correct problems in multiple systems already in operation. Also, the cost of V&V will be higher since similar V&V activities will have to be performed for each application system having mission or safety-critical functions.

Accomplishments

The research team created an initial high-level framework for performing V&V within reuse-based software engineering, by adding V&V activities to the STARS Two Life-Cycle Model. A working group at the Reuse ‘96 Workshop proposed revisions and added details to this framework. The group also considered how the new domain-level and transition-level tasks would impact the scope and level of the traditional application-level tasks

Future Plans

In order to continue developing the framework for performing V&V within reuse based software engineering, the following must be completed:

- Criteria determined for identifying domains where V&V is appropriate
- Prerequisites and inputs/outputs for the domain and transition-level V&V tasks specified
- Methods and tools to perform the domain engineering V&V tasks developed

Refinement of the framework will occur when experiments are conducted in applying V&V within critical domains. The research team will continue to interact with other groups involved with software reuse, including the Software Reuse Subgroup of the NASA Software Working Group, the Reuse Issues Action Team, the WVHTC Foundation, and the WVU Reusable Software Research Group.

Related Publications

2-4

Point(s) of Contact

Edward A. Addy
(edward.addy@ivv.nasa.gov)

Software Optimization and Reuse Technology

Objective
To serve as a vehicle for transitioning model-based, domain-specific software reuse technology to selected NASA technical centers.
To nurture the development of reuse knowledge, techniques and their adoption across NASA.
To leverage previous and on-going work performed by both NASA and the Department of Defense (DoD) in the areas of systematic software reuse.

Approach
There are two distinct but complementary tasks that comprise the SORT Program:

1. SORT DE — assessing the need for reuse; developing and applying reuse knowledge and techniques to various NASA domains to produce reuse assets
2. Technology Transfer (TT) — disseminating and teaching information obtained from the SORT Domain Engineering effort and other reuse efforts; assisting in the adoption of reuse techniques within NASA

SORT will identify and evaluate candidate NASA domains in which a need for reuse efforts exists. This necessarily involves detailed analysis of the chosen domain(s) and modeling of requirements and architectures. SORT will then leverage and promulgate the concepts and techniques of successful model-based domain-specific reuse efforts. Both of these efforts are part of an overall strategy and plan to transfer the necessary technologies to NASA centers for adoption.

Significance
By designing software for reuse in appropriate domains, software productivity, quality, and reliability can be increased while the cost and development time is decreased.

Accomplishments
One emphasis of the SORT Effort is to support DE activities on selected NASA domains. The SORT Team has been working with three Domains of Interest:

- Flight Furnaces at Marshall Space Flight Center (MSFC)
- WTCSE at LaRC
- Mission Operations System (MOS) at GSFC, JPL, and Johnson Space Center (JSC).

Future Plans
The SORT Team plans to continue performing DE in appropriate programs and transferring reuse technology to those programs.

Point(s) of Contact
Greg Blaney
(greg.blaney@ivv.nasa.gov)

Langley Research Center Wind Tunnel Control Systems Environment

Objective

To continue working on the Wind Tunnel Control System Environment (WTCSE) by initiating a Domain Design effort based on the Software Requirements Specification (SRS) completed in Phase 1 of the Software Optimization and Reuse Technology (SORT) activity with Langley Research Center (LaRC).

Approach

The Hatley/Pirbhai *Strategies for Real Time System Specification* was used as the basis for the Domain Design effort. As a result of the Domain Design effort, the Software Design Specification (SDS) was developed. The document conforms to NASA DID-P300. The SDS provides detailed requirements and design information for the WTCSE.

Significance

By generating the SDS for LaRC, SORT has provided LaRC a basis to evaluate the use of the Experimental Physics and Industrial Control System in the LaRC wind-tunnel automation environment. The SDS along with the SRS provide thoroughly documented specifications and designs which a contractor can follow. This promotes common design of future systems improving quality and decreasing maintenance costs by accommodating reuse within the component areas. Prior to this effort, contractors were assuming requirements and architectural data.

Accomplishments

An SDS was produced that represents the WTCSE's reusable common architecture, based upon the Hatley/Pirbhai *Strategies for Real Time System Specification* notation*.*

Future Plans

The SORT Team will assist the wind tunnel community by applying software reuse technology as requested in conjunction with a larger effort by the community to consolidate all wind tunnels in the US.

Point(s) of Contact

Greg Blaney
(greg.blaney@ivv.nasa.gov)

Marshall Space Flight Center Flight Furnace Interface Environment

Objective

To design software for the Space Shuttle Furnace Facility (SSFF) to be flown on the ISS.

Approach

SORT has approached the reuse initiative at MSFC based upon certain challenges faced by the MSFC personnel in developing a Furnace Facility for the yet to be implemented ISS. These personnel are required to predict the necessary interface software and hardware for eight types of furnaces, many of which are still conceptual. Their goal is to design the software with the intent of minimizing changes as new furnaces are developed and flown within the SSFF.

Significance

A reuse initiative, augmented with a DE approach, offers an alternative to this individual negotiation approach. By gathering requirements across all furnace types, modeling this information and extracting the common and variable functionality derived from the requirements, a DE approach could provide a more generic view of requirements necessary for any flight furnace facility to support the operation of multiple furnaces. These requirements would describe a Flight Furnace Interface Environment domain.

A DE effort has the added benefit of separating the functional software requirements from the hardware requirements. This should also facilitate future software modifications when needed and reduce software modifications when hardware is updated.

Accomplishments

A domain scoping and analysis effort was performed that produced a Domain Scoping Report. The report identified the necessary generic functional interface to accommodate the various types of flight furnaces to be flown in the ISS.

Future Plans

Analysis to produce a high level Requirements Model, will be completed by January 31, 1997. It will represent the functions necessary to interface the SSFF with any future furnace requirement.

Point(s) of Contact

Greg Blaney
(greg.blaney@ivv.nasa.gov)

Mission Operations Systems

Objective

To introduce the concept of reuse at NASA centers and assist in identifying reusable assets within and across the domain of MOS.

Approach

SORT is currently analyzing the super-domain of MOS at the NASA various MOS centers. The NASA centers have been identified in the following manner with their respective MOS domain:

- JSC for Manned Mission Operation System
- JPL for Deep Space Mission Operation System
- GSFC for Low Earth Orbit Mission Operation System

The SORT team will accomplish this by:

- Assessing common functions/activities performed by each MOS
- Identifying variations in functions, which support mission specific needs
- Defining a reasonable sub-domain under MOS, which shows high reuse potential
- Assessing existing and future requirements for the defined sub-domain
- Defining a generic MOS architecture which highlights the reusable assets

Significance

This domain analysis has the potential to identify areas where reuse technology should and could be applied. It will promote and allow sharing of best practices between the centers.

Work in Progress

SORT has made opportunistic contacts at GSFC and JPL and is receiving various documentation related to the super-domain and related projects.

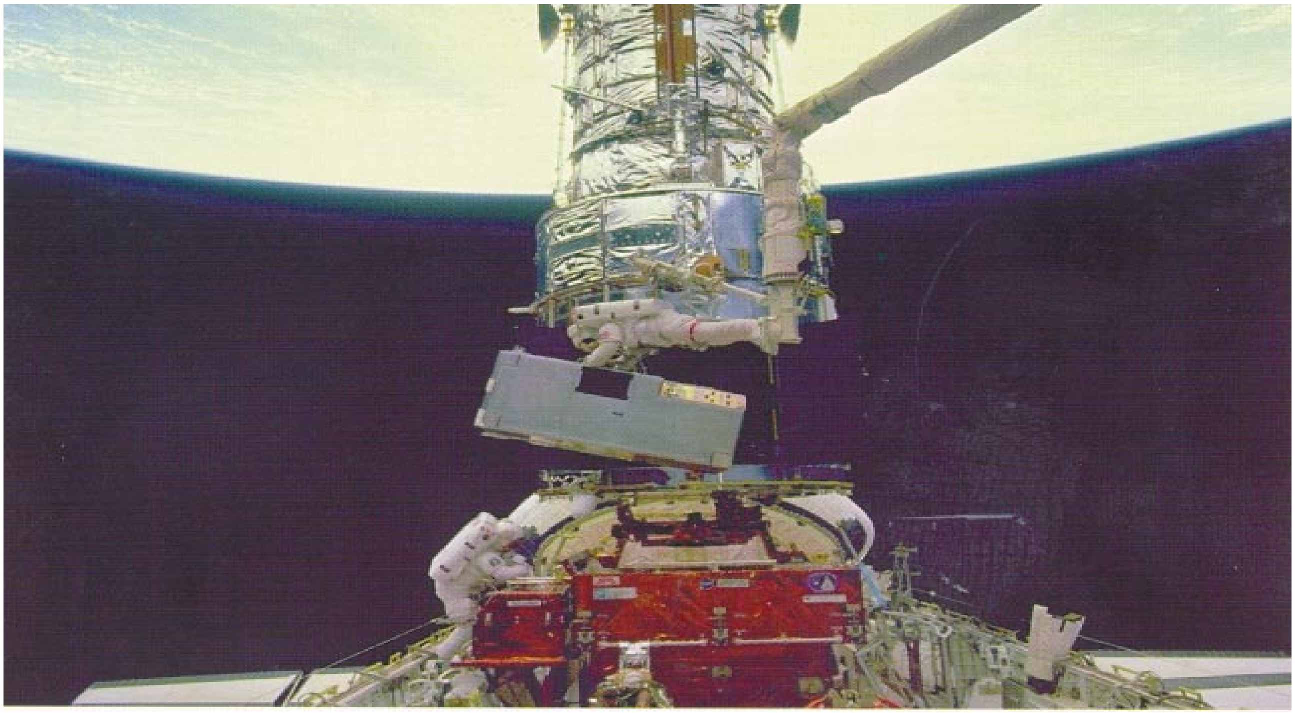
The SORT Team will continue working to introduce the concepts of reuse in the MOS domain.

Related Publications

81

Point(s) of Contact

Greg Blaney
(greg.blaney@ivv.nasa.gov)



Hubble Space Telescope

Reusable Objects Software Environment

Objective

To establish a Verifiable Development Technique for OO, reuse-based reengineering.

Approach

This technique will be developed and applied by performing IV&V on the Reusable Objects Software Environment (ROSE) Core Trajectory (RCT) project at JSC in the Mission Operations Directorate. The RCT project will deliver the trajectory sub-system infrastructure and a Vector Operations (VECOPS) application to the MCC work-station environment. The RCT project has a set of development processes and a Software Quality Plan (negotiated with the two customer organizations for the project, Flight Design and Dynamics and the Mission Control Center) which identifies the current standards required to be met by the project. The initiative funds will be used to augment the project team with individuals skilled in the areas of IV&V and real-time software testing. They will provide lessons-learned to improve the development processes and on IV&V of reuse-based reengineering.

Significance

The Mission Operations Directorate has designated the real-time trajectory software as safety-critical. As such, IV&V is required prior to use supporting a mission. However, the shuttle orbit domain also contains non-critical software for which IV&V is not required. Because both critical and non-critical applications will be built from the reusable components, IV&V should be performed on all the components, as well as the resulting applications. This initiative will demonstrate the benefits of integrating IV&V within the reuse-based lifecycle. This way, IV&V can be performed on all the domain components in a cost-effective manner.

By using OO software engineering for the common orbit functionality, and taking advantage of domain-specific reuse, the legacy on-orbit subsystem will be reduced in size by approximately 70%. In addition to there being fewer lines of code to maintain, emphasizing quality engineering practices will result in code which is less complex than the systems, better documented and thus substantially less expensive to maintain.

Accomplishments

1. Traced application (called VECOPS) support request to VECOPS requirements specification and to VECOPS design specification
2. Traced application support request to simulation domain subsystem requirements specification
3. Traced application requirements specification to simulation domain design specification
4. Performed independent testing of VECOPS application and domain components
5. Calculating McCabe line coverage metrics on application and domain
6. Completed the following initiative documents:
   - White paper on V&V
   - Draft of V&V Plan
   - Software Test Plan
   - Interim Traceability Reports

Future Plans

A report will be issued in February on the technique used to perform IV&V on reuse-based reengineering along with the results of that effort.

Point(s) of Contact

Kathryn Kemp
(kathryn.kemp@ivv.nasa.gov)

Software Analysis

Software Assurance Technology Center

Evaluation of Autonomous Spacecraft-Operations Technologies

Independent Verification and Validation Issues in Achieving High Reliability and Safety in Critical Control System Software

Software Engineering Evaluation System

Object Oriented Formal Methods

Software Structured Approach

Formal Lightweight Approach to Validation of Requirements Specifications

Case Study of DoLILU Issue Tracking Reports

Verifiable Design of an Artificial Neural Network System

Software Assurance Technology Center

Objective

To serve as a NASA-wide resource, with the goal of measurably improving the quality of the software developed for and by NASA.

Approach

As a means to accomplish the improvement of NASA software assurance capabilities, Software Assurance Technology Center (SATC) conducts programs in four areas:

1. Standards and guidance for software processes
2. Software measurements and metrics
3. Development of Tools and Techniques for Software Assurance
4. Outreach and project support

Significance

SATC tools have been applied to projects at GSFC. One of the projects estimated that it saved $50 to $100K by using the SATC tool.

Accomplishments

Guidance and Standards - SATC and previous efforts have been principal providers of software assurance related guidance and standards documents for NASA and industry use.

Software Measurement and Metrics - SATC has done extensive work in software metrics including:

- Developed a software quality model that includes both process effectiveness and product quality measures
- Developed an approach to defining a software metrics program, based on the quality model and the Goal-Question-Metric paradigm
- Obtained support to conduct research on optimization of quality and cost using OO metrics. SATC is funding a proposal from TSU for continuation of the research.

Tools That Support Software Assurance - SATC does a limited amount of development of tools to support software assurance.

- Reusable Software Management Plan - The “reusable” software management plan provides text that fills in all of the elements of the plan. The provided text can be used as is, modified, or replaced to fit the needs of an individual project. At GSFC, the SATC plan and tool were used by the EOS AM, PM, and Chemistry flights projects to develop their project software management plans.
- Data Collection - The SATC Staff has worked with Code 520 to develop an on-line data collection system. The system, Metrics Examination Reporting and Interpretation, allows data to be directly entered by the individuals and eliminates paperwork and data entry. Reports from the system are reviewed by SATC personnel and then sent electronically to project managers. This system not only will effectively support 520, but will be a SATC tool to support other organizations.

Outreach and Project Support - SATC has three ways to reach out to the software community:

- By direct project support - This currently includes monthly reporting of requirements traceability and defect tracking for the EOSDIS project; ongoing analysis of metrics data for the Software and Automated Systems Branch at GSFC; code analysis for the Landsat 7 project; and code/ problem report analysis of the Moderate Resolution Imaging Spectrometer science software project.
- By papers, tutorials, and technical reports
- By maintaining a World Wide Web (WWW) page. This information includes NASA policies, standards, guidebooks, papers, and reports.

Future Plans

An area of increased emphasis for SATC is to increase the number of NASA software development projects with which it works. Thereby directly transferring improved assurance related software technology to the projects and into practice. The Research Team is emphasizing the application of software metrics for quality assurance and risk mitigation. SATC now manages metrics programs for some GSFC projects and is seeking new ones.

Related Publications

11, 27, 28, 64, 84, 94, 95, 97-100

Point(s) of Contact

Larry Hyatt
(lhyatt@gsfc.nasa.gov)

Cynthia Calhoun
(cynthia.calhoun@ivv.nasa.gov)

Evaluation of Autonomous Spacecraft-Operations Technologies

Objective

To evaluate commercially available and advanced techniques and tools needed for validating autonomy technologies, with special emphasis on mission-level coordination activities such as spacecraft activity, planning, and sequence generation.

Approach

The approach taken for this initiative includes the following steps:

- Define critical characteristics and performance requirements of both onboard and ground autonomous operations, for better spacecraft ground requirements allocation and coordination
- Coordinate with JPL New Millennium and Discovery Mission for their technology requirements of autonomous sequence planning and generation processes
- Identify characteristics of the tools needed for validating the onboard autonomous spacecraft-activity planning and generation processes
- Survey commercially available tools that meet criteria defined: acquire, install, and evaluate two or more of these tools
- Identify and evaluate advanced techniques that meet the criteria defined; and/or complement the tools evaluated

Subsequent activities include the following:

- Refine the characteristics of the tools needed for validating the automated sequence planning and generation subsystem as identified in Phase I
- Perform feasibility study of the techniques and tools identified by demonstrating them in a New Millennium testbed

Significance

The vision for low-cost spacecraft operations is to provide an unattended operations environment, whereby no operators are visibly interfering with, or attending to, the link between Payload input and Payload user. This minimalist view requires autonomy technologies for the end-to-end infrastructure (i.e., for the spacecraft system and ground system). More specifically, autonomous spacecraft operations require that the spacecraft and instruments have the attributes of self-calibration, self-health-monitoring, and failure recovery (through data filtering, analysis, and diagnosis), and information synthesis.

Qualification of V&V tools must be accomplished in order to enable the infusion of new autonomy technologies into JPL’s spaceflight hardware and software, and into autonomous mission operations. Without the concurrent validation and development effort, establishing the reliability of onboard autonomous spacecraft activity planning and generation applications may be uncertain until long after their initial use. Pro-active and timely insertion of such V&V tools will decrease the risk of inadequate testing, while facilitating spiral development and contributing to a higher quality product.

Accomplishments

Two tool demonstration sessions to identify tools critical to the Deep Space (DS)-1 Autonomy software, took place successfully. Understanding and discussion of the purpose and relative values of some tools were the objectives of these demonstrations.

The project will follow-up with the testing team on DS-1, and will expand and tailor some of the V&V tool concepts demonstrated in support of the testing and demonstration of upcoming releases of the DS-1 autonomy software.

A presentation on the evaluation of V & V tools and concepts for autonomous spacecraft operations was developed. The presentation focused on the benefits of “on-board plan-execution” monitoring tools. The benefits of such tools fall into three areas:

- Improved testing
- Improved ground-based non-real-time health and status monitoring
- Improved “emergency-based” uplink validation

Related Publications

107

Point(s) of Contact

John Hinkle
(john.hinkle@ivv.nasa.gov)

Verification and Validation of Object Modeling Technique Models

Objective

To work with industry to develop tools that:

- Make OO technology easier and safer to use enhancing the safety, reliability, and cost-effectiveness of all NASA software developed using OO methods
- Aid in the V&V of OO models during the early life cycle phases of requirements analysis and design, and thus enhance the probability of identifying errors and misconceptions early in the life cycle, when they are less costly to correct
- Make OO technology easier and safer to use also facilitate the transfer of object technology into more NASA software projects

Approach

The Object Modeling Technique (OMT) method is still under revision, because of a few relatively weak areas, such as integration of the Object, Dynamic, and Functional models and, in general, verification and validation of the OMT models. This initiative will add an animation capability to the Dynamic model that will provide users with tool support to help test their Dynamic models. Additionally, this initiative will simulate the interaction of the object classes by animating Object Interaction Diagrams, to be added to the second release of the OMT method and will be in the Unified Modeling Language. Finally, some constraint checking will be added to the Object, Dynamic, and Functional models in Paradigm Plus (a tool that supports the OMT method), which will provide consistency and integration checking across the three sets of models. These capabilities can be used during the Analysis phase to simulate the requirements or at any other point during the development to test the models. These capabilities will be tested on NASA projects and by groups at JSC, Langley, and JPL, and refined accordingly.

Significance

Tools that make OO technology easier and safer to use can enhance the safety, reliability, and cost-effectiveness of all software for NASA strategic enterprises, and will also facilitate the transfer of object technology into NASA software projects.
This initiative is developing enhanced methods and tool support for V&V of OMT models. These enhanced methods will support testing of OO analysis and design models, making error detection during the analysis and design phases easier for the software developer and the testing specialists. These objectives will be provided, in part, by allowing simulation of planned system behavior.

Accomplishments

Accomplishments to date include the following:

- Developed test case models
- Created Tool Concept and Specification Document
- Trained lead developer in Java programming language
- Developed prototypes to demonstrate proofs of concept in five major areas:
  - Run time commands
  - Simulation of analysis models (animation-assisted simulations)
  - Interactive (user-driven) simulations
  - Batch-driven simulations
  - Diagram switching (model integration and visualization)
- Identified potential test sites
- Installed Paradigm Plus, an OO modeling tool, on development platform

Future Plans

- Mature the prototypes to a presentable state
- Demonstrate the tool (and as-of-yet unprototyped concepts) at NASA and at commercial tool vendors
- Collect evaluations from demo attendees and project developers on tool’s perceived usefulness
- Document FY96 results in a report describing the technology and its project benefits to NASA plus recommendation on whether NASA should invest in this technology in the near future

Point(s) of Contact

Charles Pitman
(cpitman@ems.jsc.nasa.gov)

Independent Verification and Validation Issues in Achieving High Reliability and Safety in Critical Control System Software

Objective

To ensure the safety of critical software systems by integrating the elements of safety analysis, reliability analysis, and metrics analysis into a comprehensive risk reduction program.

Approach

Safety Analysis - The National Research Council (NRC) report recommends the use of Fault Tree Analysis (FTA) to identify any hazards that may exist in the STS software. This technique could also be used to model hazards in other systems developed by NASA (e.g., system fault protection and recovery software for the Mars Pathfinder and Mars Global Surveyor spacecraft), as well as systems proposed by other agencies (e.g., tele-medicine systems under consideration by the US Army and the Advanced Research Project Agency).

Reliability Analysis - Currently available software reliability models are applied in order to:

- Predict time to next failure and remaining failures for areas of the software that the FTA has identified as containing hazards
- Use the reliability predictions to identify software that should receive priority attention for FTA due to relatively low reliability predictions

Metrics Analysis - The use of metrics as early indicators of reliability will be examined. Like FTA, the use of metrics is designed to provide early indicators of reliability so that corrective action can be taken early in the life of the software. The use of metrics and reliability will be integrated by utilizing the metrics validation methodology from the IEEE Standard 1061 Standard for a Software Quality Metrics Methodology. The purpose of the validation will be to identify those metrics (e.g., the set proposed by the GSFC Software Assurance Technology Center) which have sufficient association with defect report counts, failure counts and time to next failure — both observed and predicted — to serve as early indicators of reliability. Specifically, the models are intended to express the rates at which faults are introduced and removed as functions of development process characteristics, product characteristics, and the number of faults already in the product. These rate expressions could then be used in a series of linked birth and death models to predict the fault content at any future time. Preliminary results over a small number of projects indicate that these rates are not dependent on the number of faults already in the product during the later development phases (e.g., detailed design, implementation), but that there may be such a dependence during earlier development phases.

This arrangement will make it possible for NASA to benefit by applying the results to improving the quality of Shuttle flight software.

As a result of previous work:

- A tremendous fault library has been created.
- Instrumentation is in place to measure the functionality of the STS system.

Significance

Risk analysis IV&V is an important elements in ensuring the safety of critical software systems. If the risk is unacceptable, steps are taken to reduce it. By definition, IV&V and risk analysis are related: IV&V can be used to reduce risk by employing inspection, testing, safety analysis, reliability analysis, and metrics analysis.

Software reliability and software safety share the goal of designing into the software the reliability and safety required to reduce risk to an acceptable level. With the release of the NRC report, it is important to develop approaches to reduce risk and increase reliability and safety that go beyond the use of reliability models. At the same time, the important contribution of these models in providing developers with confidence in the operational readiness of the software must be retained.

Accomplishments

- Developed Ada source code analyzer and tool for tracing evolution of source code
- Completed preliminary analysis of source code structure (changes in relative complexity) for six delivered builds of the CDS
- Began characterization of Cassini development process with respect to Cost Containment Model 2.0 criteria
- Began collecting and analyzing data from NASA Scatterometer Science Data System
- Obtained information to commence risk/software reliability analysis for STS

Point(s) of Contact

Ann Patterson-Hine
(apatterson-hine@mail.arc.nasa.gov)

Software Engineering Evaluation System

Objective

To determine the suitability of the Software Engineering and Evaluation System (SEES) for use by NASA and to develop a reusable Generic Evaluation Methodology (GEM) applicable to the evaluation of IV&V techniques.

Approach

This initiative was a joint center effort that included LaRC, JSC, GSFC, and later MSFC. It was proposed originally in FY93 in order to establish a generic approach to evaluating IV&V methodologies. The evaluation approach was to be used as the basis for evaluating various IV&V methodologies being considered for use at SS&T. Once the generic evaluation approach was developed it would be used to evaluate the Army developed SEES which was being used on missile command projects. The SEES methodology was to be evaluated by using a test bed approach under controlled conditions and by applying the methodology to several pilot projects in order to evaluate its effectiveness in a real project environments.

Significance

The project assessed the effectiveness of SEES as applied to NASA projects, the effectiveness of the GEM as an evaluation tool, and the effectiveness of the data collected in support of the GEM. The evaluation project consisted of three independent projects to apply SEES to NASA software development activities, and a central activity to provide technical management, data collection and analysis, and to develop the Evaluation Project Final Report.

Accomplishments

- Trained NASA and contractor staff in the SEES IV&V methodology
- Developed the Generalized Evaluation Methodology for IV&V methods, based on pilot projects
- Developed experimental plan for the formal, statistical, evaluation of an IV&V process
- Collected data about the application of the SEES IV&V Methodology
- Completed a quasi-experiment based on the formal plan
- Analyzed of the data collected

Future Plans

The final report is in its final stages and will be completed during the second quarter FY97.

Related Publications

25, 30, 45, 46, 79, 85, 86, 90, 105, 106, 116

Point(s) of Contact

Kathryn Kemp
(kathryn.kemp@ivv.nasa.gov)

Object Oriented Formal Methods

Objective

To integrate the two disciplines in order to explore system testing and testability issues.

To define and carry out pilot projects using potions of existing large-scale space programs.

To reduce formal methods to state-of-the-practice on programs of national importance.

Approach

The explicit structuring techniques and graphical notation contributed by OMT, complemented by the verification of key properties and behaviors contrib uted by formal methods, will provide a mechanism for designing testability into the system early in the lifecycle. This will:

- Support testing throughout the lifecycle, from the requirements through the implementation phase
- Focus the testing process at every stage on crucial properties and behaviors
- Provide the necessary traceability between system specification and test artifacts.

This initiative will apply Formal Methods (FM) techniques to large-scale projects where OO requirements and specification techniques are used. The study will build on previously successful efforts to transfer FM technology to selected NASA programs. Stanford Research Institute (SRI) International's PVS was the FM toolset used in these prior studies and will form the basis of this new proposed study.

Significance

Many large-scale software projects are now using the OMT or a similar method to add structure and rigor to the early lifecycle activities of requirements analysis and high-level design. Formal methods offer complementary techniques that likewise add precision to the early lifecycle phases. While OMT offers explicit structuring mechanisms and graphical modeling techniques that appeal to designers, its ability to capture the semantics of the system under study and to reason about proposed system properties and behavior is very limited. Conversely, formal methods offer theoretically powerful modeling and analysis capabilities, but lack built-in structuring concepts so that analysts must construct their models from first principles. An integration of these two disciplines would amplify the effectiveness of analysts and designers, yielding higher quality systems with fewer residual defects passed on to later lifecycle phases.

Accomplishments

The LaRC contribution on this multi center initiative was completed in FY96. This included the volume II guidebook, and writing and presenting the following papers:

Related Publications

24, 31, 32, 41, 42, 88

Point(s) of Contact

Rick Butler
(rwb@qirl6.larc.nasa.gov)

Kathryn Kemp
(k.kemp@ivv.nasa.gov)

Software Structured Approach

Objective

To integrate Formal Methods and Analytical Verification (FM/AV) into a full set of verification, validation, modeling, and design techniques for critical software subsystems.

To act as a catalyst in providing transfer materials for NASA projects beginning to use these techniques.

Approach

FM/AV is a significant set of widely researched techniques and tools based on logic and mathematical models for the purpose of verifying software requirements, design, and code. Prior work by members of the current research team demonstrated the effectiveness of formal techniques on NASA spacecraft flight software. One of the needs in developing and assuring critical software, is a set of integrated formal analysis models which maintain fidelity with one another and with development products. These models will enable more effective verification of software subsystems throughout their development & maintenance life spans. The techniques are currently not linked and integrated to provide leverage with other techniques. In addition this study will develop a case study report to provide guidance on how to integrate OO design methods within the FM/AV frame work.

Included under the transfer material portion of this Center Initiative is 1) the development of training materials to support the recently developed guidebook series, 2) a WWW information center to provide assistance to adopters of FM/AV

Significance

Software requirements and design have been a significant quality issue for critical NASA systems. Studies have indicated that the most hazardous software safety errors result from of requirements discrepancies or weak interface specifications. FM/AV has been demonstrated to bring rigor and structure to these early lifecycle products by reducing ambiguities, creating high level logic models, and employing deductive techniques. During the piloting of FM/AV techniques on several NASA space flight software systems, two additional areas were uncovered which will improve the quality of highly critical software systems. The first is the need to develop formal specification library components to more quickly and accurately model flight software systems. The second is to integrate FM/AV into the other development and verification techniques which are already in place on critical software projects, thus increasing the leverage that can be gained to ensure reliability. This task is designed to directly address both of these problems.

Accomplishments

Several case studies have been conducted on NASA spacecraft flight software subsystems. Results indicated the usefulness of an FM/AV approach. Forty-six issues/questions were found that escaped traditional analysis and testing. Issues were:

- One incorrect logic
- One circular reasoning
- One redundant test
- Five confusing notations of logic
- Three type mismatches
- Three misspellings
- Three confusing notations
- Twenty-nine clarifications

A two volume set of NASA Guidebooks has been developed to aid projects transistioning to the FM/AV method of assuring and verifying critical systems.

Future Plans

Future plans include advancing FM/AV techniques. The first direction,during FY 97 and FY 98, is the development of software flight verification components to enable faster and reduced cost analysis of requirements and design.

The second direction is the tailoring and piloting of integrated FM/AV techniques on advanced flight software systems in cooperation with the New Millennium Project (NMP). The FM/AV demonstrations conducted to date have had relatively standard software architectures. The NMP is testing architecture which includes far greater automation and built in intelligence than previous spacecrafts.

Related Publications

1, 21, 22, 40-42, 53-56, 59-60, 62, 69

Point(s) of Contact

John C. Kelly
(john.c.kelly@jpl.nasa.gov)

Kathryn Kemp
(kathryn.kemp@ivv.nasa.gov)

Formal Lightweight Approaches to Validation of Requirements Specifications

Objective

To explore the use of formal methods for finding errors in specifications.

Approach

The Research Team is conducting a series of case studies, applying formal modeling techniques to 'live' projects, so that the results can be fed back into the project in time to be of added value. Where possible, the case study is a response to a real need on an existing project, for example where an additional level of assurance of the correctness of the requirements is needed, over and above that obtainable through existing methods. The Research Team is applying a range of different formal methods, including Software Cost Reduction (SCR), PVS, and the model checker Software Process Improvement Network (SPIN). In each case the Research Team is examining the amount of effort required to apply the method, and the types of benefit gained. This evaluation is mainly qualitative, as the baseline metrics do not exist for detailed quantitative comparisons.

Significance

Formal methods have not yet gained widespread acceptance by software practitioners. Part of the problem has been an emphasis on adopting a baseline a formal specification, from which to prove that design and implementation are correct. The Research Team's work has demonstrated that a more realistic approach is to use formal methods for small pieces of modeling, to answer questions that cannot be addressed in other ways.

Accomplishments

The research team has conducted a case study of the requirements for Fault Detection Isolation and Recovery (FDIR) for the ISS, using SCR and SPIN.

The study demonstrated that a great deal of effort was needed to formalize the requirements, especially in obtaining a correct interpretation of the original prose. A number of errors were detected during the normalization process, and these were reported to the IV&V team.

Future Plans

The Research Team is investigating two key issues:

1. If formal methods are applied early in the requirements phase, when requirements are still relatively unstable, then effort is required to keep the formal models up to date. The Research Team is exploring techniques to facilitate this.
2. By asking several people to formalize the same requirements and comparing the results, the Research Team reveal ambiguities and inconsistencies in the original specification. This approach also helps to check that the derived model is the intended interpretation.

Related Publications

13, 15, 36, 38, 39

Point(s) of Contact

Steve Easterbrook
(steve.easterbrook@ivv.nasa.gov)

Case Study of the DoLILU Issue Tracking Reports

Objective

To determine if IV&V is an effective technique for identifying critical problems during early phases of the software development lifecycle.

Approach

Day of Launch I Load Updates (DoLILU) Issue Tracking Report(s) (DITR) act as the nuts and bolts for the success of IV&V. One hundred and nine DITR were prepared and the phases of the life cycle to which each DITR was applicable were identified. Although the DITR framework can be fit into as few as four and as many as eight phases, the software life cycle was divided into six standard phases to make this study compatible with other IV&V effectiveness studies. When problems from each DITR were recorded separately, a total of 695 problems were identified. There were nine DoLILU software components associated with the 695 problems. Each problem was identified with a DoLILU software component and a particular phase in the life cycle.

Significance

One component (Day of Launch I-Load Verification Data Table (DIVDT)) accounted for almost 50% of the problems. Another component accounted for over 15%. Three components accounted for 7.6-8.5% each, and the remaining four components accounted for less than 4% each. Coding was started on DIVDT before the requirements were written. The developer started writing code with only a vague understanding of the full requirements. This resulted in most of the errors being shifted from requirements and design to the coding phase. The same scenario of coding without requirement and design documents is common for the other components of this system. This is the explanation for the large number of errors found in phase 4.

Accomplishments

The small number of errors found during phase 5 (late in the lifecycle) indicates that IV&V was an effective technique in finding errors that escaped the developers. From the data gathered on DoLILU I, the findings seem to coincide with earlier studies that suggest that IV&V can be a cost effective technique for identifying problems early in the software development lifecycle if it is performed from the beginning of the software lifecycle. The Research Team's findings also seem to coincide with previous findings that IV&V is less cost effective if it is performed only at later phases (especially after phase 3).

Future Plans

Using the DoLILU I case study as a spring board, data from DoLILU II will be studied to determine if the trends found in DoLILU I continued. The Research Team is also investigating correlation studies to determine cause-effect relationships between IV&V activities and trends in issue reporting.

Related Publications

74

Point(s) of Contact

John R. Callahan
(john.callahan@ivv.nasa.gov)

Verifiable Design of an Artificial Neural Network System

Objective

To develop a systematic methodology for formal verification and testing of Artificial Neural Network (ANN)-based control system.

Approach

Essential to the verification and testing of any engineering system is the availability of an accurate model and methods of testing the model against the real system. The design of ANN based system so far has not included these verification steps due to the lack of knowledge of the system at the time of design. The Research Team's approach is divided into two parts. For the first part the Research Team is working to develop a method to extract simple control rules from the trained ANN system. The Research Team achieves this through an Adaptive Network Based Fuzzy Inference System (ANFIS). The second part of the Research Team's effort is to build an accurate system model based on domain knowledge modified by rules extracted from the real ANN system. First, domain knowledge(incomplete and inaccurate) are used to build a coarse system model. Then the Research Team instantiates and compares it with the rules that are extracted from the real system. These two steps iterate until the modified model is fully instantiated and agrees with the rules that are extracted from the real system. After an accurate system model is generated, the Research Team can then apply formal verification and testing methods that are developed for well-defined systems.

Significance

Recent years have seen increased applications of ANN based control systems. Examples include the ANN controlled propulsion system developed by NASA Dryden facility, ANN based auto-pilot system developed by Professor Napolitano's group and US Army's smart bomb project. This new paradigm of system design has brought many unanswered questions with regard to stability and safety issues. Systems developed using ANNs must be subject to the same level of verification and testing effort that has been status quo for conventional system. This is especially true for safety critical systems such as those used in aviation.

Accomplishments

The Research Team's main effort has included the selection of a proper system modeling method and an efficient rule-extracting algorithm for the ANNs. The Research team has chosen the SCR four-variable model for its simplicity and wide use. The Research team has experimented with two rule extracting algorithms which have been well acclaimed. One is the NeuralRule algorithm developed by Setiono [3](National University of Singapore) and the other is the ANFIS tool developed by Jang[1](UC Berkeley). The Research Team adopted the latter for its ability to extract piece-wise line rules from arbitrary control surface. The Research Team is in the process of applying it seriously to the real ANN based auto-pilot developed by Professor Napolitano's group[2](West Virginia University). Due to the high dimension of their system, the Research Team need to enhance the ANFIS tool so it can deal with real world problems.

Future Plans

The Research Team is planning to enhance the ANFIS tool so that it can handle multi-dimensional rule extraction. After the enhancement the Research Team will be able to apply it to extract control rules from the real ANN based auto-pilot developed by Professor Napolitano's group.

References

1. Jang, R, *Anfis: Adaptive-Network-based Fuzzy Inference System,* IEEE Transactions on Systems, Man, and Cybernetics, 23(3):665-685, 1993.
2. Napolitano, R, and M. Kincheloe, *On-line Learning Neural Network Controllers for Autopilot Systems,* in 95' American Institute of Aeronautics & Astronautics Guidance Navigation and Control Conference, Baltimore, MD, August 1995.
3. Setiono, and H. Liu, *Symbolic Representation of Neural Networks,* IEEE Computer Magazine, pages 71-77, 1996.

Related Publications

109-111

Point(s) of Contact

John R. Callahan
(john.callahan@ivv.nasa.gov)

Technology Transfer and Community Service

Science and Engineering Apprentice Program

Hands on Science

Software Assurance Training

Technology Transfer and Other SORT Activities

Remote Sensing Public Access Center

Software Working Group

Software Process Improvement Technology Transfer

Technology Transfer and Community Service

Objective

To increase technology awareness within various communities by acting as an information center.
To connect the industry's needs to a center of excellence within NASA.
To make available the proven technology developed by NASA.
To return the technology developed by NASA to the public.

Approach

The TT effort is made of three focus areas:

- Provide technical training for NASA Safety & Mission Assurance (S&MA) engineers in software assurance
- Coordinate efforts with the high schools and grade schools to further the interest in science and engineering
- Facilitate making NASA technology available to industry

The initial interface, or the "outreach" portion of the TT activity will be done by the Development Office. Code IT will provide the link to a NASA Center activity.

Significance

The transfer of Software Process Improvement to the local industry and other NASA Centers is progressing via several routes. The local industries participate in the application of IV&V as contractors/ subs at SS&T. Weekly meetings with the IV&V research staff and the practitioners have been established to exchange needs, ideas, etc. These meetings effectuate TT by guiding the research path and giving practitioners new tools as they are developed for pilot test.
TT impacts NASA as well. The export of software technology directly assists NASA development of new software and interfacing with industry helps NASA understand the direction of technical growth.

Accomplishments

The TT program has worked a verbal agreement with the West Virginia Development Office. This office is an arm of the state government which works with the West Virginia industry to help them grow and develop new products and jobs. The efforts implemented by TT have impacted composite materials in the Thyises aircraft and EOS related unmanned aircraft.

Future Plans

Future efforts will include working with software companies in the high tech corridor to develop commercial applications from tools/processes, developed for or by NASA; and allied with the regional TT office in Pittsburgh. This effort will serve to commercialize software tools/processes and assist with other NASA outreach within West Virginia.

Point(s) of Contact

John Griggs
(john.griggs@ivv.nasa.gov)

Science and Engineering Apprentice Program

Objective

To give selected students the opportunity to perform meaningful engineering work in the form of a project which can be completed in eight weeks, under the guidance of a mentor who is working here at SS&T. To give the student a taste of the life in an engineering environment, complete with the writing of a report on the project and the results.

Approach

The Science Engineering Apprentice Program (SEAP) is a summer intern program of eight weeks duration. Students who can apply are in their sophomore or junior year of high school at the time of application. At the completion of the internship, a trip to Washington DC is given as a reward and an opportunity to present the results of the project to a peer group that has worked on similar projects.

Significance

It is anticipated that the United States will be faced with a shortage of scientists and engineers in many areas. SEAP was established to combat this problem by generating interest in these disciplines among students that are academically able, but unmotivated due to lack of exposure and knowledge.

Accomplishments

During the summer of 1995, SEAP had a pilot intern program with five students. For the summer of 1996, the number was increased to ten, with the various programs within SS&T providing mentors and projects. The experience indicated that ten is the ideal scope of the program. The staff was sufficiently challenged in providing the interns with meaningful projects.

Both years have been adjudged successful. The board of education and school personnel have strong praise for the pogrom and its results.

Future Plans

1. Continue the program at ten interns per summer.
2. Encourage the interns to use their reports in science fairs and college scholarship applications.
3. Assist (starting in three years) those who continue through college in finding local employment where their chosen field is supportive of the NASA mission here.

Point(s) of Contact

John Griggs
(john.griggs@ivv.nasa.gov)

Year	Intern	Project Title/Subject
1995	Brian K. Bennett	NASA Computer Communications
1995	Susan Davis	Code QV: The Tactical Plan
1995	Thomas B. Miller	Interface Analysis of the International Space Station Guidance, Navigation, and Control using SRS and ICD Documentation
1995	Martin Felix Padula	Library Database Creation
1995	Erika L Peters	Software Optimization and Reuse Technology (SORT)
1996	Brian K. Bennett	Weather Facsimile
1996	Bill Bradley	Automated Web-based Electronic Forms
1996	Nicholas Butcher	Science Engineering and Technology Assessments (SETA) project
1996	Mikaelah Cianfrocca	SORT Domain Engineering
1996	Richard Bradley Harter	Development of a Graphic User Interface for the Software Improvement Management Optimization Network
1996	Gevony Blair Laughlin	Web-based Hypertext Environment for Requirements Evolution
1996	Zach Moore	Making Hypertext Markup Language (HTML) documents accessible via Hypertext Transfer Protocol (HTTP)
1996	John Murphy	SORT Domain Engineering
1996	David Schwartz	Insight into the Software Life-Cycle and Independent Verification and Validation (IV&V) Through a Small Tool Development Project
1996	Rebecca L. Wright	Groundtruth Analysis

NASA Software Systems & Technology Facility SEAP Interns

Hands on Science

Objective

To provide opportunities for children to engage in science as a pleasurable activity.
To draw parents into interaction with the child’s science education through take home materials.

Approach

Hands on Science is an after school class where students perform science experiments at their level while a volunteer teacher explains the principals behind the results being obtained. The apparatus used is at the grade level of the students.
While the program covers K through 6, it has been decided to gear the program to grades 4 and 5. Emphasis has also been placed on the rural areas of Marion county for this effort, trying to interest the children in learning more math and science as they go through the remainder of their education.

Significance

The program generates enthusiasm by giving the experiment apparatus to the child to take home and share with parents and friends. This interaction is designed to increase the child’s interest in the activity.

Accomplishments

Twelve kits were placed in the elementary schools in the fall of 1995 and the spring of 1996. Eighteen kits (Twelve were NASA provided and six came from grant money provided to the school system) were placed in the schools in the fall of 1996. Each kit contains eleven student sets and one instructor set of materials for eight experiments. During the fall of 1996, 198 students participated in the program. One school had a waiting list of 22 students.
There are three certified trainers for preparing the in-classroom volunteers in the county. All of the trainers are prepared to trainers volunteers in adjacent counties as funds become available.

Future Plans

1. Continue the current program making it grow to a larger audience in surrounding counties
2. Continue providing motivational support to encourage the interested students to follow a math/science path through high school

Point(s) of Contact

John Griggs
(john.griggs@ivv.nasa.gov)

Software Assurance Training

Objective

To raise the qualification level of the S&MA staff to journeyman, hands on.
To make the training available to contractors, at their cost.

Approach

The implementation of the software assurance training program, Phase I of the effort as documented in the Code Q plan, and the Fairmont State College (FSC) Science Applications International Company proposal are underway. These efforts are first validating the need, via comparison of the Software Working Group (SWG) need definition and the Code Q Professional Development Initiative (PDI) need matrix. The training plan is being geared to the journeyman level S&MA engineer, realizing that the individuals are degree holding engineers, but not in an S&MA discipline.

Significance

This effort will prepare the NASA S&MA staff to perform, hands on, the assurance analyses on the software developed internally to the Agency. This is necessary because of the reduction in support contractors who have traditionally performed these tasks. Most of the S&MA engineers in the Agency are engineers in the traditional fields, i.e. Electrical, Mechanical, etc. This training is, in part, a cross-training into the assurance discipline.

Accomplishments

A partial survey of available training was performed under the guidance of the PDI group, and will also be factored into the training plan.
The training plan, including curriculum, was completed by December 1996. The first report, a comparison of the two matrices mentioned above, was presented at the SWG training sub-group on December 5, 1996, at GSFC.

Future Plans

A study of the possible certification of the graduates of the training is under way by FSC. Graduate credit for the majority of the courses is also under study, and may be provided in conjunction with WVU engineering schools.
With the aproval of the completed training plan near the end of January 1997, Phase II of the effort will commence. Phase II will first map existing courses, modules, etc. to the needs analysis in the plan and then define the delta. Where there is no available course, one will be developed and Beta tested in this phase. Any modifications, additions, or deletions to the overall curriculum will be made, and preparations made for Phase III, the operational instruction of the S&MA cadre.

Point(s) of Contact

John Griggs
(john.griggs@ivv.nasa.gov)

Technology Transfer and Other SORT Activities

Objective
To disseminate software reuse technology.

Approach
SORT has worked to share reuse technology by attending/sponsoring conferences and seminars, presenting papers, creating web pages, and heading various subgroups within NASA.

Significance
Reuse technology is vital to the future of software production and assurance.

Accomplishments
Conferences, Seminars, and Workshops

- European Space Agency (ESA) 1996 Product Assurance Symposium and Software Product Assurance Workshop (19-21 March 1996, Noordwijk, The Netherlands)
- Object Oriented Rapid Application Development Workshop (11-15 November 1996, Toronto, Canada)
- Defense Information Systems Agency Domain Engineering Process Course (11-15 November 1996, McLean, VA) To Be determined (TBD)
- Software Engineering Institute (SEI) Feature Oriented Domain Analysis and Domain Engineering Course (18-20 November 1996) TB

Papers Presented

- *Domain Engineering - An Enabling Technology for Software Product Assurance*, ESA 1996 Product Assurance Symposium and Software Product Assurance Workshop (19-21 March 1996, Noordwijk, The Netherlands)
- *Reusing Information on Human Functions to Improve Architecture-based System Design*, Software Technology Conference (STC) 1996 (22-28 April 1996, Salt Lake City, Utah)

Workshops Held or Cosponsored by SORT

- Workshops held or cosponsored by SORT
- A NASA Focus on Software Reuse (23-27 September 1996, George Mason University)
- SORT Technology Transfer Workshop (Date TBD, NASA Ames Research Center)

Web Development

- SORT web page developed - “http://sort.ivv.nasa.gov”
- Developed web based inventory system (Reuse Registration Form) for Reuse Subgroup of the Software Working Group - “http://sort.ivv.nasa.gov/reuse_rf.htm”

Reuse Subgroup Lead

- SORT has been named as the “Execution Arm” for the NASA SWG Reuse Subgroup.
- Coordinated the development of the *NASA Reuse Subgroup Charter,* which was signed off by the whole subgroup.
- Reuse Registration Form developed to inventory Reuse programs/efforts internal and external to NASA.
- Coordination of bi-weekly meeting to include but not limited to: setting up agenda, information gathering, documentation, teleconferencing management, and reporting of subgroup’s activities to the SWG.

Future Plans
SORT will continue efforts to make reuse technology available to industry, government agencies, and the public.

Related Publications
43, 68, 75, 102

Point(s) of Contact
Greg Blaney
(greg.blaney@ivv.nasa.gov)

Remote Sensing Public Access Center

Objective

To support NASA's Information Infrastructure Technology and Applications (IITA) Project teams and their activities.

To increase public access, via the Internet, to space observations of the earth, our solar system, and the universe beyond, through WWW sites and outreach activities.

Approach

The Remote Sensing Public Access Center (RSPAC) researches and develops Internet technology tools that benefit IITA projects, establishes mechanisms for collaboration and communication among the projects and provides assistance and expertise in earth ans space science, system administration, and WWW technologies.

RSPAC increases IITA visibility and showcases NASA data through the *Observaorium* Website, presentations at conferences, appearances at regional education facilities, and both traditional and Internet-related publicity.

Accomplishments

- RSPAC has produced a suite of Internet technology tools for the IITA projects. These include *The Inquisitor*, a customized software tool that monitors and records Website visits; and *The Validator*, an HTML syntax verifier. *The Inquisitor* is in use by more than 30% of the IITA projects.
- RSPAC provides services to the IITA projects. These include Web Site Mirroring, Website Test and Evaluation, and Graphics and Multimedia asistance.
- RSPAC developed various mechanisms for the integration and exchange of knowledge among IITA projects. These include the *Developer's Workshop*, a Website created for the projects; the *PI Bulletin*, a monthly newsletter; and specialized electronic mailing lists.
- RSPAC provides the infrastructure to support reliable WWW servers. Our servers have sustained over 21 million hits with an average of 1.7 hits per month. RSPAC currently hosts nine mirror sites for IITA projects.
- *The Observatorium*, RSPAC's primary public Website, showcases the IITA projects and NASA earth and space science data. It has received 1.5 million hits from over 52,000 visitors, and was selected as a NASA Cool Site of the Week.
- RSPAC promoted the IITA projects by exhibiting at thirteen national conferences and presenting at four regional education workshops.
- RSPAC produced *Exploring the Internet* with NASA, an interactive CD-ROM tutorial that showcases NASA Internet science and teaches Internet basics.

Point(s) of Contact

Stratis Kakadelis
(stratis.kakadel@rspac.ivv.nasa.gov)

Software Working Group

Objective
To evaluate, advise, and promote the advancement of software engineering, management, development, and assurance across NASA.

Approach
The SWG is an Agency-wide software advocate and coordinating body that is responsible for addressing software related issues throughout NASA. The SWG currently has members from NASA Headquarters and all NASA Centers. The SWG Charter outlines objectives, functions, and roles and responsibilities of the SWG.

Significance
Software driven programs are vital to the success of NASA missions. The SWG will, as stated in the SWG Charter, work to:

- Focus and integrate the software programs throughout NASA
- Define and recommend the goals of the NASA Software Strategic Plan
- Provide guidance for all programs containing software
- Ensure that available software processes and procedures are disseminated to all NASA programs

The SWG is responsible for performing the following functions:

- Ensure the goals and strategies of the NASA Software Strategic Plan are supportive of the NASA Strategic Plan and mission
- Recommend implementation strategies and priorities consistent with the needs and requirements of the NASA programs and projects
- Guide the full implementation of the NASA Software Strategic Plan
- Recommend and provide technical support for special studies and assessments in support of the NASA Software Strategic Plan

Accomplishments
In response to this charter, the group developed a NASA Software Strategic Plan, which addresses NASA's software vision, mission, and goals and strategies to be implemented throughout NASA. The NASA Software Strategic Plan was finalized and signed by the SWG members on July 13, 1995.

Future Plans
The SWG meets at least twice a year to work on software initiatives that support the goals outlined in the NASA Software Strategic Plan:

Goal 1: Implement and integrate software engineering processes into systems engineering on NASA programs. Software engineering, assurance and management products and services will be integral to the planning, development, risk management, and implementation processes of the programs and operations contained within NASA's Strategic Enterprises and Functions.

Goal 2: Transfer software technology. Innovative software technologies, processes, and techniques will be transfer into the NASA system/software engineering approach augmented through focused research. NASA advanced software technology will be transferred.

Goal 3: Continually improve NASA's software engineering processes to produce measured improvements in the cost and the quality of software developed for and by NASA.

Goal 4: Maintain Agency capabilities in software technology. The NASA work-force will have the necessary skills to effectively manage software projects and apply software technology.

Point(s) of Contact
Kathryn Kemp
(kathryn.kemp@ivv.nasa.gov)

Michele Choban
(michele.choban@ivv.nasa.gov)

http://www.ivv.nasa.gov/SWG

Software Process Improvement Technology Transfer

Objective

To provide a step by step approach along with question/answer support for NASA organizations just beginning a software process improvement program. To cover the expansion into training materials of the information contained in the guidebooks and subsequent piloting of this training.

Approach

In order to build upon previous LaRC and GSFC experience in producing training, the courses developed will be deployed at least twice with time allowed for repackaging between offerings. Videos will be made after the courses mature. The proposed training courses are:

1. “Establishing a Software Measurement Program” for practitioners
2. “Software Process and Product Improvement within NASA” (potentially 2 courses: high level and more detailed level)
3. “Software Management” for practitioners

This new proposal envisions the packaging and deployment of further training courses at different levels (center management, project management, development and maintenance staff) focused on the same concept of establishing a software improvement and measurement program within a local organization.

The final step of the NASA Software Process Improvement Approach is to package successful software development and maintenance processes and management approaches for use by subsequent projects.

Significance

This initiative supports the needs of NASA software organizations to manage their projects and improve the software processes used. This activity strongly supports the NASA Software Program and Software Strategic Plan goals of transferring successful NASA software processes, of promoting the use of software metrics for mission success, and of increasing expertise in software management. This initiative will establish a NASA experience based training program for software engineering professionals.

Accomplishments

The “Software Measurement Guidebook”, the “Software Process Improvement Guidebook”, and the “Software Management Guidebook” were produced in FY95 under a Code Q Software Engineering Program Center Initiative. In addition, a one hour briefing on “Measurement for Managers” was developed.

Future Plans

The course materials, instructors’ notes, and videos will be turned over to the Office of Human Resources and Education to widen the scope of these courses

Related Publications

9, 10, 76, 77, 80, 82, 91-93, 96, 117, 118

Point(s) of Contact

Rose Pajerski
(rpajersk@pop500.gsfc.nasa.gov)

Kathryn Kemp
(kathryn.kemp@ivv.nasa.gov)

Flight Software process Definition and Implementation

Process Definition for Rapid Development of Software

Software Risk Management Guidebook and Training

NASA Software Assessment Procedure and Guidebook

Guidebook for Safety Critical Software-Analysis and Development

Flight Software Process Definition and Implementation

Objective

To define, document, and implement a standard baseline process for use in developing Lewis space flight software.

To use this baseline as the foundation for continuous process improvement.

Approach

The effort, will leverage off previous Code Q funded work in this area (at Langley and Goddard), both by attempting to reuse/tailor the results of those efforts and by taking advantage of past experiences. The first step in this effort will be to fully understand the processes and approaches developed at LaRC and GSFC. This will be accomplished via interviews and/or surveys. Following completion of these activities, a thorough assessment of all information obtained will be made, and a plan developed which details the appropriate manner in which to proceed. Once completed, the Process Documentation Phase will begin. This phase will consist of the iterative process of documenting, submitting for review, and integrating comments received on the defined process. Upon completion of this phase, final baseline documents will have been created, and implementation of the defined process can begin. During the Implementation Phase, pilot projects will be identified to begin using the documented approach, collecting the specified metrics, etc. In addition, training will be developed and utilized for the purpose of infusing this approach back into the organization.

Significance

Currently at Lewis Research Center (LeRC), there is no standard, defined process in place for developing flight software. This means that each project reinvents the software process to be used, with varying degrees of effectiveness, and with little agreement across projects or personnel as to the appropriate activities or procedures to be used at any given stage. As the LeRC software projects have grown in size, complexity, and number, this problem has compounded itself to the point where both the software community and LeRC management have recognized the need for one standard process to be defined as a baseline for use by all. By defining and documenting the desired software process in the manner described above, the cumulative individual knowledge which currently exists at LeRC will be translated into institutional knowledge, thereby improving the overall effectiveness, efficiency, and quality of the software engineering process at LeRC. Furthermore, by leveraging off previously funded Code Q work in this area, TT is being utilized, and the innovative processes/techniques which provide these benefits are being developed at reduced overall cost.

Accomplishments

The Software Engineering Process Group, which will lead software process improvement effort, has been established and is beginning to implement approach specified in the project plan.

Five LeRC flight projects underwent process capability assessment during Program/Project Management Initiative Software Process Improvement pilot course, and project process strengths and areas for improvement were identified.

Point(s) of Contact

Cynthia Calhoun
(cynthia.calhoun@ivv.nasa.gov)

Process Definition for Rapid Development of Software

Objective

To define and document generic, rigorous processes and guidelines for rapid development, integrated design, and verification of flight software.

To document the practical tailoring and application of these processes to project work in the JSC/ Aeroscience & Flight Mechanics Division Guidance Navigation & Control (GN&C) Rapid Development Laboratory (RDL).

To enable the dissemination and further application of the rapid development techniques to other NASA Centers and Programs, and to the commercial sector by building upon work currently being performed in the GN&C RDL.

Approach

- Generate a lexicon of rapid development terminology
- Define and document a draft standard for rapid software development technical and management processes with special attention on processes for verification of auto-coded software
- Define and document process performance metrics for rapid software development and verification
- Correlate the draft standard to the SEI CMM
- Demonstrate the application of the draft standard by gathering associated performance metrics from JSC/GN&C RDL projects

Significance

Many of today's system problems are so complex that advanced software development approaches are necessary if they are to be solved in a timely, useful, and cost effective manner. Technology is advancing so quickly that a system that meets requirements frozen at some historical moment may be obsolete before it is delivered. All indications are that the complexity of the problems faced will continue to increase and exist in environments of constant change, and thus may require these new methodologies if they are to be successfully solved. The process, guidelines, metrics, and tailoring examples will provide a basis for meeting these challenges to software development.

Accomplishments

- Completed the initial structuring of a GN&C Deorbit Flight Software Demonstration Project using the concepts in the "Guidelines for the Rapid Development of Software Systems
- Completed the initial work to install a Metrics Program for the GN&C Deorbit Flight Software Demonstration Project

Point(s) of Contact

David Petri
(dpetri@gp903.jsc.nasa.gov)

Bill Jackson
(bill.jackson@ivv.nasa.gov)

Software Risk Management Guidebook and Training

Objective

To document and provide training on processes for identifying, analyzing, communicating, and averting software project, product, and process risks.

Approach

The guidebook, developed by SEI, will provide instructions on how to perform specific risk management techniques and examples to help the reader understand the concepts involved. A short training course will be developed to provide instruction on the concepts involved and exercises to gain the necessary skills in applying risk management techniques. The course will be given at LeRC.

Significance

OSMA has previously funded the development of a full life-cycle software process guidebook for small space-flight projects at LaRC. The process defined in the guidebook has a strong emphasis on risk management and the ability to respond quickly to problems or deviations in the project plan. It contains risk management techniques and examples specific to the flight software domain. Under this initiative, LaRC would use those previously developed products and lessons learned as a foundation for developing a generic NASA Software Risk management Guidebook and associated training.

Accomplishments

A “Continuous Risk Management Course” was developed geared to the NASA domain. A draft of the course contents were delivered for review in November.

A detailed case study, based on the NASA domain, has been completed by civil servants and delivered to the contractor to use as the basis for the course examples and exercise.

Future Plans

As part of this initiative, the risk management database portion will be upgraded to Windows 95 and Access 7.0. This database will be made available via the WWW.

The final course materials will be delivered in the third quarter of FY97.

A dry run to NASA subgroup members on Continuos Risk management course is scheduled at LeRC March 17-19.

The final guidebook, course materials, instructors’ notes, and video tape of the course presentation will be delivered to the Office of Human Resources and Education (Code F) to widen the transfer of this technology across NASA.

Point(s) of Contact

Pat Schuler
(m.p.schuler@larc.nasa.gov)

Siamak Yassini
(siamak.yassini@ivv.nas.gov)

NASA Software Assessment Procedure and Guidebook

Objective

To enhance the software self assessment process that was initiated at LaRC in FY95 to include measurements activities which have been developed by the GSFC and documented in the “Software Measurement Guidebook”. Under this initiative, LaRC will be responsible for integrating the significant attributes of software measurement, into the draft Self-Assessment Guidebook.

Approach

The initiative will provide guidance to aid in the identification and elimination of deficiencies in the software engineering process of individual organizations. The primary responsibility of those organizations is to provide software products in support of NASA’s strategic enterprises and improve the quality of software products developed by NASA. In addition, by incorporating measurements/metrics into the overall self-assessment procedure, individual organizations will record their current baseline as well as quantitatively measure process improvement over time. The self-assessment reports and associated action plans will be provided as examples of recommendations for corrective action in particular deficiencies.

Significance

By incorporating measurements/metrics into the overall self-assessment procedure, individual organizations will record their current baseline as well as quantitatively measure process improvement over time. The Self-Assessment Reports and associated Action Plans will be provided for use as examples of recommendations of corrective action for particular deficiencies and examples of plans to implement corrective action.

Accomplishments

In support of NASA-wide software measurement collection, a low cost database to capture core software metrics is being developed. The PC hosted software metrics database initial version has been delivered. Functional demonstrations of the database have been provided.

Point(s) of Contact

James Watson
(j.f.watson@larc.nasa.gov)

Kathryn Kemp
(kathryn.kemp@ivv.nasa.gov)

Guidebook for Safety Critical Software - Analyses and Development

Objective

To advance the state of the art for NASA in the area of safety analysis for software (including firmware). This initiative will develop:

1. A standardized, comprehensive, straight forward approach to building safe software
2. The necessary guidance for those performing the analyses
3. A document to aid management in understanding the time and cost necessary to provide a certain level of software safety

Approach

The entire software life cycle will be covered from the view of what analyses are available and which are appropriate for each life cycle phase. A complete "how to" approach is the essential element of this guidebook. This is to assure understanding of the techniques and a standard approach across NASA for conducting them.

Several of approaches and techniques, once put together, will be used on a space experiment, Combustion Module (CM)-1 to prove out some of the newer methods.

Significance

With the many space experiments, critical aeronautics work, and Expendable Launch Vehicles which must be assured; a standardized, comprehensive, straight forward approach to building safe software will provide the necessary guidance for those performing the analyses. In addition, it will help management understand the time and cost necessary to provide a certain level of software safety.

Accomplishments

The Software Safety Guidebook initiative was successively completed last Spring/early summer. The guidebook provides a standard approach to setting requirements, and examining and determining software safety. It was generated from work in progress, namely the Software Safety Standard and an early version of Space Station Software (S/W) Fault Analysis Plan (unbaselined), IEEE standards, Mil-specs, Space Shuttle Program 30309, JPL Handbook, and lessons learned from several projects. It is accessible via the SS&T Website.

Point(s) of Contact

Kathryn Kemp
(kathryn.kemp@ivv.nasa.gov)

Software Engineering Process Guidebook

Objective

To document a repeatable, measurable, and tailorable software engineering process with proven procedures and tools.
To have the guidebook address all the Key Process Areas of the SEI CMM in levels 2 and 3.
To ensure the process is measurable as a basis for process improvement.

Approach

The guidebook is currently under development, and it will contain five volumes:

- Introduction
- Management
- Development
- Configuration Management
- Process Improvement

Each volume contains a process that defines how the activities performed under that discipline will interact. Each process activity will have a suggested method associated with it. References to external documentation will be made where possible. There will be an appendix which contains all the forms referenced in the guidebook. An additional appendix will relate the available software tools to the activities in which they may be used. The guidebook will be applicable to the LaRC Software Engineering and Analysis Laboratory (SEAL) software applications/ domains (i.e. flight, mission operations, data processing, and ground support equipment including simulators). However, much of the guidebook will be applicable to domains outside SEAL.

Significance

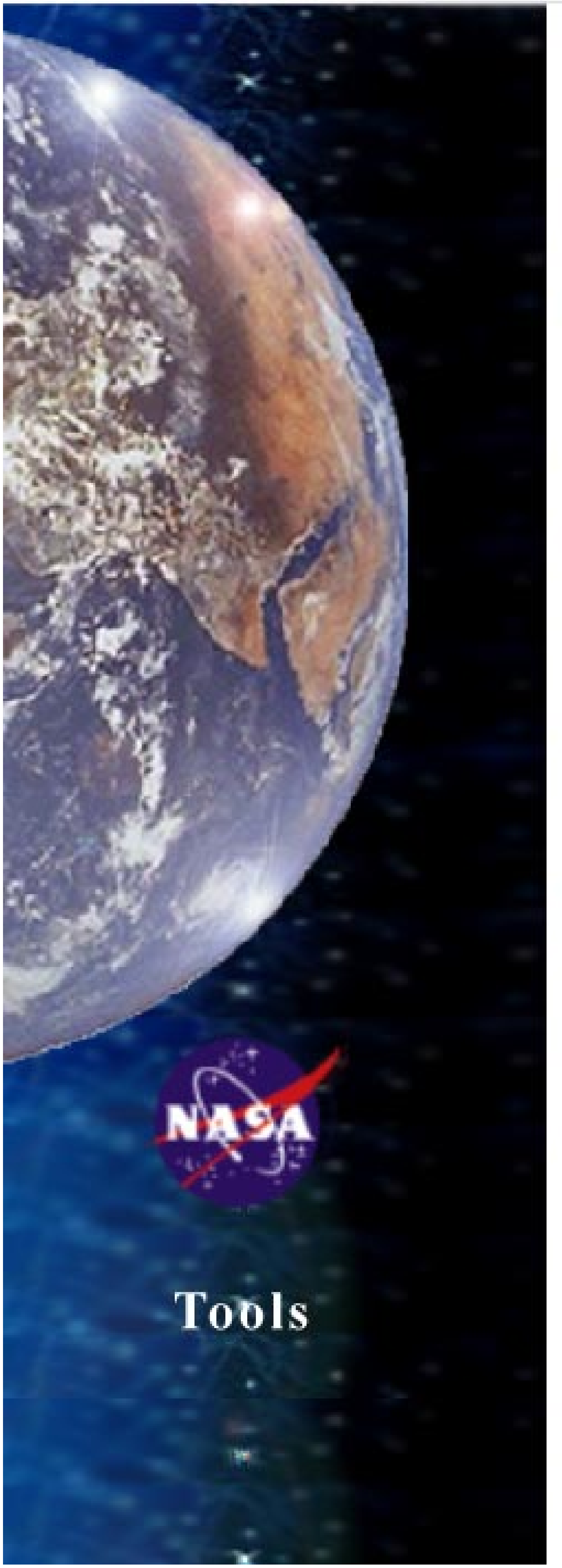
By infusing improved technologies, deficiencies will be reduced in the software engineering processes of organizations whose responsibility is to provide software products. Therefore, the overall quality of the products will be improved. Areas of concentration include: Configuration Management, Quality Assurance, Software Measurement for use in process improvement and more accurate budgeting and scheduling. Existing NASA measurement, software engineering, and process improvement guidebooks are heavily leveraged under this activity.

Accomplishments

The guidebook documents a repeatable, measurable, and tailorable software engineering process. Draft versions of the Introduction, Management and Process Improvement volumes have been delivered. The final hypertexted version of all of the volumes was delivered in November 1996.

Point(s) of Contact

Kathryn Kemp
(kathryn.kemp@ivv.nasa.gov)



Application of Formal Testing to Reliable Multicast Software

A Web-based Hypertext Environment for Requirements Evolution

Automated Network of Software Engineering Resources

Quantitative Software Methods

WWW-based Integrated Software Metrics Environment

System Safety Techniques for Automated Fault Protection/Software Safety Techniques

Application of Dynamic Flowgraph Techniques for Safety Analysis and Testing of Space Systems Software

Software Improvement and Management Optimization Network

Application of Formal Testing to Reliable Multicast Software

Objective

To develop and apply formal testing tools and approaches to development and maintenance of reliable multi-cast software.

Approach

It is important for testing and verification methods to use real world systems as an application testbed. It is also very important that this testbed be demonstrated to be non-trivial. Reliable multi-cast software is such a testbed application. The formal testing approach uses the development model in conjunction with formal models to help guide test cases selection and analysis. This project has developed formal models of the Reliable Multi-cast Protocol (RMP) and used them to help develop test suites and testing frameworks. Recently, another reliable multicast approach, Scaleable Reliable Multicast (SRM) has been implemented in a generic C++ framework (Generic SRM or GSRM). By examining and evaluating several reliable multi-cast approaches and applying formal testing methods and tools to them, it is believed that the approach and the tools can be improved and “honed” for use on other distributed applications.

Significance

Reliable broadcast and multi-cast protocols will play major roles in the development of future large-scale data systems. For example, such protocols will be needed to maintain coherent copies of data at multiple sites in an efficient manner. Due to the complexity and criticality of such protocols, it is important that they be developed and researched in an atmosphere where correctness and verification are integral parts of the process. Due to the non-trivial nature of the problem domain and the commercial impact of reliable multi-cast, the use of formal testing has an opportunity to have a large impact on software development practices.

Accomplishments

The research team has developed RMP, formal state models of RMP, and a suite of test cases for RMP. In the process of developing this, the team has been able to maintain and demonstrate high fidelity between the RMP implementation, formal state model, and testing model. As a secondary benefit, the tools used in these procedures are in the process of being generalized to the more general problems addressed. A toolkit is being developed and supported in an effort to make this directly applicable to other software projects. This toolkit is being supported and developed by the Software Research Laboratory (SRL) under the name of Software Research Laboratory Testing (SRLT) Toolkit.

Future Plans

The research team will continue to develop and analyze the RMP and GSRM implementations as well as investigate other reliable multi-cast approaches in an effort to further develop formal testing approaches, methods, and tools. The SRLT toolkit will continue to evolve and be supported. New analysis methods, such as simulation/emulation of WAN dynamics, will be investigated as well.

Related Publications

17-19, 65-67, 112, 113

Point(s) of Contact

John R. Callahan
(john.callahan@ivv.nasa.gov)

A Web-based Hypertext Environment for Requirements Evolution

Objective

To tackle inconsistency management problems in large specifications.

Approach

Specification sets are inconsistent for most of their lifetimes, because they are constantly being edited. The challenge is to detect inconsistencies and to keep track of them, so that unresolved inconsistencies do not lead to incorrect decisions, and resolved inconsistencies stay resolved.

The Research Team is developing a set of scenarios to illustrate the kinds of coordination problems that occur. They are using these to direct the next stage of the research work: tool building. The Research Team is plan to build a set of web-based tools to manage the relationships between chunks of specification. These tools will be introduced incrementally on real projects. The aim is to generate an initial set of tools that both offer added value to their users and allow us to collect more data about where coordination problems occur. In the first set of tools, the Research Team will provide limited functionality for recording meta-data about specifications, and for reviewing/annotating them.

Significance

Many projects are now using the hypertext and the web as a way of organizing project documentation. However, no model yet exists for exploiting the linking capability of hypertext. Existing requirements traceability tools model high-level relationships between specifications, but are based on coarse-grained process models. The Web-based Hypertext Environment for Requirements Evolution project offers a way of modeling detailed dependencies between specifications, so that inconsistencies do not propagate through the documentation.

Accomplishments

The Research Team has interviewed a number of IV&V analysts, and produced scenarios to describe interactions between IV&V and development teams. The Research Team has used these scenarios to pinpoint problem areas, and has generated a list of priority areas where tool development is likely to have the highest impact. The Research Team has also started work on porting an existing specification editing toolset to Java, to be used as the foundation for a web-based specification annotation system.

Future Plans

The Research Team will continue to develop a toolset for editing specifications. The goal is to build a web based specification review system, in which various types of specification documents can be annotated and edited via the web. The tool will keep track of changes made, and improve the ability of the developers to keep track of how changes made by others affect them.

Related Publications

33-35, 37

Point(s) of Contact

Steve Easterbrook
(steve.easterbrook@ivv.nasa.gov)

Automated Network of Software Engineering Resources

Objective

To provide projects and managers with a suite of WEB-based tools that will enhance their ability to identify, capture, assess, track, and control project data.

Approach

The Automated Network of Software Engineering Resources (ANSWER) is a collection of automated software applications designed to provide collection, management, and dissemination of software management and engineering information/data. ANSWER applications incorporate the tools, techniques, methodologies, metrics, and results obtained from NASA improvement projects and research. Other tools and applications may be integrated into the final ANSWER product as requirements dictate. ANSWER features will address the collection, maintenance, and reporting of features including:

- Issue Tracking
- Requirements Engineering
- Problem Reporting and Corrective Action
- Best practices
- Experience Database
- Software Development Processes and Products
- Software Development Indicators (Metrics)
- Software engineering tools, techniques, and methodologies

Significance

The many advantages that such an automated effort would bring to a development group include the following:

- Overcoming the geographical barrier: By using a Web-Based technology one can access another resource anywhere on the Internet. This benefits the software teams who can access the tool from any location.
- Overcoming the communication barrier: Encourages collaborative software problem solving.
- Problem Solving: Software Managers can effectively track the progress of their work group by using performance measures built into the tool itself. The different views the tool supports would give a better perspective of the problem database, would help to handle large amounts of detail, and would help to pinpoint problems early and suggest timely action,
- Data Continuity: changes to the database are reflected in all views. Continuous flow of data between users and managers helps coordinate the work products of many people who work on a common software project.
- Software Assurance: Software Assurance Organizations will have access to a system that will provide them with technical and objective evidence necessary to support review and acceptance activities.

Accomplishments

The ANSWER prototype provides on-line information about the development process and design ideas for the collaborative web environment in order to gather feedback from potential user groups. The following documents have been included:

- Technical Approaches
- System Specification
- Software Requirements Specification
- User Survey

The ANSWER prototype has identified five tools:

1. Labor Utilization Tool - Provides for submission and review of electronic timesheet data
2. Project Expenses Tool - Provides for submission and review of project costs
3. Project Action Item Tool - Tracks issues throughout the project lifecycle
4. System Trouble Report Tool - Tracks software problems discovered during testing phases
5. System Change Request Tool - Tracks software enhancements to be performed

Point(s) of Contact

Rhonda Fitz
(rhonda.fitz@ivv.nasa.gov)

Quantitative Software Methods

Objective

To provide a toolset that will:

1. Be useful to measure the necessity and completeness of a safety critical Change Request (CR) or a patch (for a matured/operational software).
2. Be useful to help the designer and tester to know where the trouble spots are and to facilitate the writing/development of more effective test cases.

Approach

This project produced a software toolset, which measures the relative complexity of SS software components, and a reliability model which will combine into a Software Reliability Assessment Toolset.

Significance

The concepts of the relative complexity are based on the assumption that the conditions which lead to software faults are identifiable as a set of measurable attributes. Hence the Research Team can use those software attributes that are associated with faults to identify regions of software code currently under development or test that are likely to contain faults. For software which is not written in HAL/S, the Research Team can use other analyzers (available in the Information Systems Directorate) to replace Tool 1 and still use Tool 2.

Accomplishments

The Software Reliability Assessment Toolset was delivered at the end of the project.

1. Tool 1 - A code analyzer which deals with High Order Assembly Language/Shuttle (HAL/S) source code. Current commercially available code analyzers are designed for the other computer languages such as FORTRAN, C, and Ada. Since the SS flight software is written in HAL/S code and is significantly different from the other languages, a HAL/S code analyzer is therefore needed and produced with this Center Initiative project.
2. Tool 2 - A tool that will measures the relationship between the software faults and their complexity domains. This tool will take the metrics generated from the first tool and use them to calculate the relative complexity of the software.

Both tools have been delivered to SS&T.

Related Publications

8, 23, 26, 29, 44, 47-52, 57, 63, 78, 83, 87, 89, 101, 103, 104, 108

Point(s) of Contact

Alice Lee
(alice.t.lee1@jsc.nasa.gov)

Kathryn Kemp
(kathryn.kemp@ivv.nasa.gov)

WWW-based Integrated Software Metrics Environment

Objective

The automated collection of software metrics and subsequent analysis of measurements as a means of managing evolutionary development and promoting improvements in software practice at the level of the individual developer much like the SEI Personal Software Process Improvement Model.

Approach

This diagram depicts the idea that issues derived from IV&V are fed back to the developer where they are used to drive development. Issues derived from development are fed forward to IV&V where they are used to drive IV&V efforts.

The identification and disposition of issues is of primary importance to a software development project. Measurement of the issues process can reveal useful data on the status of a project for management purposes. These measures include number of open issues, number of closed issues, rate of issue closure, average time to close issues, and so on. Such metrics can be used to compare projects. More importantly, changes in value over the lifetime of a project for some of these measures give an indication of the maturity of a project.

By automating the collection of measures such as these, the Research Team can collect a large amount of data, without any intrusion to the project. The Research Team is exploring the use of instrumented tools that support the issues process, while collecting data on the process automatically.

Accomplishments

The Research Team developed WWW-based Integrated Software metrics Environment (WISE) in order to demonstrate a “proof of concept” regarding collection and analysis of software metrics.

Significance

Some of the questions that this research will answer are:

- Does WISE help predict workflow and/or release date?
- Does WISE focus attention on problem areas?
- Does WISE help in understanding where a project stands in relation to its schedule?
- Does WISE help Personal Software Process Improvement?
- Does WISE fit Goal/ Question/ Metric and other models?
- Does WISE help predict &/or plot risk?
- Does WISE help prioritize work?
- Does WISE allow the dynamic rescheduling of workload?
- Does WISE allow issues to be grouped for easier understanding?
- Is WISE dangerous to the software development process if the feedback loop is tightened too much?

Future Plans

The Research Team is looking for a real project on which to use WISE to answer some of these questions. The Research Team will continue to use student projects to test theories and improve understanding of workflow and schedule dynamics

Related Publications

12, 16

Point(s) of Contact

John R. Callahan
(john.callahan@ivv.nasa.gov)

System Safety Techniques for Autonomous Fault Protection/Software Safety Techniques

Objective

To advance the state of software and system safety techniques to keep pace with changes in NASA software development processes and applications.

Approach

The initiative has two main components:

1. To update software safety techniques needed to support the rapid development of autonomous spacecraft having innovative architectures
2. To integrate some existing safety techniques which have been used with success separately on both software and hardware into the system engineering processes.

The products from the first component will be an initial and final case study evaluating appropriate software safety techniques for the DS-1 spacecraft of the NMP and ongoing “mini-deliverables” of preliminary results to the NMP development team. The products from the second component will be a method for merging software and hardware safety analyses, demonstrated on the critical system fault protection functions of DS-2, DS-3, and/or the Mars projects.

Significance

This initiative will advance the use of software and system safety techniques within NASA. It will update current practices to meet some of NASA’s changing needs and integrate improved software safety techniques into the development environment on NMP. The difficulty of integrating software engineering techniques with the results of similar analyses on the hardware adds risk, especially for critical, highly coupled functions such as system fault protection. This initiative will improve the integration of the software and hardware analysis results by documenting and demonstrating a repeatable method.

Accomplishments

The primary accomplishments to date on this initiative are in two areas.

Generic Monitor Mapping - The data and functions from a previously developed formal model (OMT and PVS) specifications of the design of a generic Cassini software monitor were used to check the requirements for the threshold and transaction monitors on DS-1. This mapping not only helped validate some current requirements for DS-1, but also identified candidate requirements for future builds of the software. The mapping also clarified requirements allocation among software components and documented constraints on the design.

Requirements Modeling - The SCR tool from the Naval Research Laboratory has been used to model the requirements for safety-critical portions of the DS-1 spacecraft. Results from the requirements modeling have been fed back to the DS-1 project. Consistent with the rapid, evolutionary development of software on DS-1 and the project’s extensive use of Internet web sites, a mini-deliverable product form was created. With this, preliminary results of work-in-progress have been logged and posted to the web pages for ready access by the development team.

Related Publications

7, 61, 114, 115

Point(s) of Contact

Ann Patterson-Hine
(apatterson-hine@mail.arc.nasa.gov)

Application of Dynamic Flowgraph Techniques for Safety Analysis and Testing of Space Systems Software

Objective

To apply Dynamic Flowgraph Methodology (DFM) to a select space experiment which has safety critical embedded software.

Approach

The application of DFM will assess the viability of using this tool in early detection of system level safety and reliability issues. It will provide knowledge of unexplored aspects of safety related software.

DFM takes into account the dynamic nature of a real time embedded system, the various states and/or modes and the transition between them. Traditional analyses mostly address only the static states. DFM uses timed fault trees to follow the software's execution paths and model its effects on the related hardware and software.

An experiement is being conducting modeling the CM-1 Structure of Flameballs at low Lewis numbers (SOFBAL) experiment (a LeRC Space Experiment) first from requirements documents and then from detailed design and code. A demonstration at each phase is to be given as well as reports.

Significance

The experiment is not only providing a safety analysis of one project, but is giving LeRC the opportunity to learn this new method first hand. It is also providing a presentation of the final finding along with a report comparing the method to the traditional methods of Safety and Reliability analysis.

If this methodology proves to be the significant leap in software safety and reliability it seems to be, NASA will want to incorporate this technique into its way of performing hazard analyses and reliability assessments of safety critical software systems

Accomplishments

- Completed the first pass on modeling the CM-1 SOFBAL experiment, analyzed created fault sequences and "prime implicants"
- Started rework of DFM model of the CM-1 SOFBAL project using design information from project
- Compared results to current NASA LeRC Safety Hazard Analysis approach

Point(s) of Contact

Ann Patterson-Hine
(apatterson-hine@mail.arc.nasa.gov)

Software Improvement Management Optimization Network

Objective

To design, integrate, implement, and maintain a management and operations network that will facilitate the exchange of management information between the SWG members and the Software Technology Division Management located at SS&T.

Approach

This initiative provided an initial suite of WEB based capabilities to enable the SWG members to more effectively communicate. The capabilities include provisions for providing:

1. Points of contact for SWG activities, issues and concerns
2. Information and management capabilities for SWG and associated Subgroup activities including meetings, telecons, and other related software events
3. Discussion groups/areas for software issues and concerns throughout NASA
4. Information about software related activities throughout NASA
5. A means to track actions assigned during SWG meetings/telecons and subgroup meetings/telecons
6. A means to review center initiative deliverables

Significance

Software Improvement Management Optimization Network (SIMON) will provide the SWG and the Center Initiative managers an internal communication forum and working area for software-related issues and concerns.

Accomplishments

SIMON was well received by the SWG and has been in use by group members as a means of communication. Primary usage has been for E-Mail and use of the threaded discussion features which are used to conduct subgroup discussions.

Future Plans

Currently, planning is underway to upgrade SIMON to the next generation of expanded capabilities. This new generation will be called the Software Working Group Information Exchange.

Point(s) of Contact

Kathryn Kemp
(kathryn.kemp@ivv.nasa.gov)

Appendix A - Acronyms

ACVC	Ada Compiler Validation Capability
ANFIS	Adaptive Network based Fuzzy Interference System
ANN	Artifical Neural Network
ANSWER	Automated Network of Software Engineering Resources
ARDB	Automated Requirements Database
CARA	Criticality Analysis and Risk Assessment
CDS	Command and Data Subsystem
CM	Combustion Model
CMM	Capability Maturity Model
COFR	Certificate of Flight Readiness
COTS	Commercial off the Shelf
CR	Change Request
DE	Domain Engineering
DFM	Dynamic Flowgraph Methodology
DITR	DoLILU Issue Tracking Report
DIVDT	Day of Launch I-Load Verification Data Table
DoD	Department of Defense
DoLILU	Day of Launch I Load Update
DS	Deep Space
EOSDIS	Earth Orbiting Data and Information System
ESA	European Space Agency
FDIR	Fault Detection Isolation and Recovery
FM	Formal Methods
FM/AV	Formal Methods and Analytical Verification
FME	Formal Methods Europe
FMSP	Formal Methods in Software Practice
FOSE	Foundations of Software Engineering
FRR	Flight Readiness Review
FSC	Fairmont State College
FTA	Fault Tree Analysis
GEM	Generic Evaluation Methodology
GFE	Government Furnished Equipment
GN&C	Guidance, Navigation & Control
GPS	Global Positioning System
GSFC	Goddard Space Flight Center
HAL/S	High Order Assembly Language/Shuttle
HSS	High Speed Simulator
IEEE	Institute of Electrical and Electronics Engineers
IITA	Information Infrastructure Technology and Appllications
ISP	Information Sharing Protocol
ISS	International Space Station
IV&V	Independent Verification and Validation
JPL	Jet Propulsion Laboratory
JSC	Johnson Space Center
LaRC	Langley Research Center
LeRC	Lewis Research Center
MCC	Mission Control Center
MEDS	Multi-Function Electronic Display System
MOS	Mission Operations Systems
MSFC	Marshall Space Flight Center
MSRA	Mission Software Readiness Assessment
NASA	National Aeronautics Space Administration
NMP	New Millennium Project

NRC	National Research Council
OI	Operational Increment
OMT	Object Modeling Technique
OO	Object Oriented
OSMA	Office of Safety and Mission Assurance
PC	Personal Computer
PCS	Portable Computer System
PDI	Professional Development Initiative
PITS	Project Issue Tracking System
PHA	Preliminary Hazard Analysis
PVS	Prototype Verification System
RCT	ROSE Core Trajectory
RDL	Rapid Development Laboratory
RMP	Reliable Multicast Protocol
ROSE	Reusable Objects Software Environment
RSPAC	Remote Sensing Public Access Center
S&MA	Safety & Mission Assurance
S/W	Software
SATC	Software Assurance Technology Center
SCR	Software Cost Reduction
SDS	Software Design Specification
SEAL	Software Engineering and Analysis Laboratory
SEAP	Science and Engineering Apprentice Program
SEES	Software Engineering Evaluation System
SEI	Software Engineering Institute
SFMEA	Software Failure Modes and Effects Analysis
SIMON	Software Improvement Management Optimization Network
SOFBAL	Structure of Flameballs at low Lewis numbers
SORT	Software Optimization and Reuse Technology
SPIN	Software Process Improvement Network
SQL	Structured Query Language
SRI	Stanford Research Institute
SRL	Software Research Laboratory
SRLT	Software Research Laboratory Testing
SRM	Scaleable Reliable Multicast
SRS	Software Requirements Specification
SS	Space Shuttle
SS&T	Software Systems and Technology
SSFF	Space Station Furnace Facility
SSME	Space Shuttle Main Engine
STARS	Software Technology for Adaptable, Reliable Software
STC	Software Technology Conference
STS	Space Transportation System
SWG	Software Working Group
SWGIE	Software Working Group Information Exchange
TBD	To Be Determined
TMDB	Test Management Database
TT	Technology Transfer
US	United States
V&V	Verification and Validation
VECOPS	Vector Operations
WAN	Wide Area Network
WISE	WWW-based Integrated Software Metrics Environment
WTCSE	Wind Tunnel Control Systems Environment
WVHTC	West Virginia High Technology Consortium
WVU	West Virginia University
WWW	World Wide Web

Appendix B - Publications

1. Abernethy, K, *Courseware in Support of Guidebook I- Managers Course Materials,* 1995.
2. Addy, E, *A Framework for Performing V&V Within Reuse-Based Software Engineering,* Symposium on Software Reusability, NASA-IVV-96-016 WVU-SRL-96-016, 1996.
3. Addy, E, *Experience Report: The Use of Functional Flows in IV&V,* 6th IFIP International Working Conference on Dependable Computing for Critical Applications, DCCA-6, NASA-IVV-96-006 WVU-SRL-96-003, 1996.
4. Addy, E, *V&V Within Reuse-Based Software Engineering,* in Proceedings for the Fifth Annual Workshop on Software Reuse Education and Training, Reuse '96, NASA-IVV-96-014 WVU-SRL-96-014, 1996.
5. Addy, E, J. Callahan, M. Prasad, & T. Montgomery, *Analysis of the Modified Real Time Executive Operating System as Used by the Cassini Command and Data Subsystem (CDS),* October 1996.
6. Addy, E, J. Callahan, M. Prasad, & T. Montgomery, *Assessment of the Scheduling and Interrupt Handling Function of Cassini Command and Data Systems,* December 1996.
7. Ampo, Yoko, & R. Lutz, *Evaluation of Software Safety Analysis Using Formal Methods,* Workshop for Foundations of Software Engineering (FOSE), Hamana-Ko, Japan, 1996.
8. *Applications of the Schneidewind S/W Reliability Model,* Johnson Space Center, September 1994.
9. *Assessing Software Engineering Technology Transfer Within NASA,* January 1995.
10. *Assessment of the Usage of Software Standards at NASA,* June 1995.
11. *Automated Analysis of Requirement Specifications,* 14th Annual Pacific Northwest Software Quality Conference, October 1996.
12. Callahan, J, *Infrastructures for Collaborative Enterprises,* AI Magazine, 17(1), 1996.
13. Callahan, J, & F. Schneider, *Specification- Based Testing using Model Checking, in* Proceedings of the 1996 SPIN Workshop, Rutgers University, 1996.
14. Callahan, J, & G. Sabolish, *A Process Improvement Model for Software Verification and Validation,* Journal of the Quality Assurance Institute, NASA-IVV-94-008 WVU-SRL-94-008, 1996.
15. Callahan, J, & J. Morrison, *Use of SCR Specifications in Verification and Validation*, in 4th Annual Workshop on Software Cost Reduction Specification Techniques, Naval Research Lab, Washington, DC, NASA-IVV-94-009 WVU-SRL-94-009, 1994.
16. Callahan, J, & S. Ramakrishnan, *Software Project Management on the World-Wide-Web,* in Proceedings, IEEE 5th Workshop on Enabling Technologies, Infrastructure for Collaborative Enterprises (WETICE'96), Workshop on Requirements Engineering in and for Networked Enterprises, Stanford, CA. NASA-IVV-96-006 WVU-SRL 96-006, June 19-21, 1996.
17. Callahan, J, & T. Montgomery, *Approaches to Verification and Validation of a Reliable Multicast Protocol, in* Proceedings of the International Symposium on Software Testing and Analysis, San Diego, CA, 1996.
18. Callahan, J, & T. Montgomery, *Decentralized Software Bus Based on IP Multi-casting,* in 3rd Workshop on Enabling IEEE Fifth Workshops on Enabling Technologies, Infrastructure for Collaborative Enterprises (WETICE'96): Morgantown, WV, NASA-IVV-94-001 WVU-SRL-94-001, 1994.
19. Callahan, J, & T. Montgomery, *Verification and Validation of a Reliable Multicast Protocol,* 2nd Annual NASA/BSC Safety Through Quality Conference, Cape Canaveral, FL, NASA-IVV-95-006 WVU-SRL-95-006, 1995.
20. Callahan, J, T. Zhou, & R. Wood, *Software Risk Management through Independent Verification and Validation*, in 4th International Conference on Software Quality (ICSQ 94), McLean, VA, NASA-IVV-94-002 WVU-SRL-94-002, 1994.
21. Cheng, Betty, & Brent Auernheimer, *Applying Formal Methods and Object-Oriented Analysis to the NASA Space Shuttle,* Michigan State University, Technical Report, MSU-CPS-94-9, 1994.
22. Cheng, Betty, & Brent Auernheimer, *Applying Formal Methods and Object-Oriented Analysis to Existing Flight Systems,* of 18th Annual Software Engineering Workshop, Greenbelt, MD, December 1993, pp. 274-282.
23. *Controlling & Predicting the Quality of Space Shuttle Software Using Metrics,* Johnson Space Center, July 1994.
24. Crow, Judith, & Ben L. DiVito, *Formalizing Space Shuttle Software Requirements,* Workshop on Formal Methods in Software Practice (FMSP '96), San Diego, CA, January 10-11, 1996.
25. Design Review & Assessment of SEES (DRATAP:3-1, DRAWKS:3-1), January 1994.
26. *A Detailed Look at the HAL/S Metric Analyzer,* Johnson Space Center, September 1995.
27. *Developing a Successful Metrics Program,* 2nd Annual Conference on Software Metrics, Washington, DC, June 1996.
28. *Developing An Effective Metrics Program,* European Space Agency Software Assurance Symposium, Reprinted Society for Software Quality Journal, March 1996.
29. *Development of Execution Profiles* of Quantitatve Software Methods*,* Johnson Space Center, May 1994.
30. *Development Process Assessment* of SEES (DPAPAP:6-1, DPAWKS: 6-1), April 1993.

31. DiVito, Ben L, *Formalizing New Navigation Requirements for NASA's Space Shuttle,* Formal Methods Europe (FME '96), Oxford, England, March 20-22, 1996.

32. DiVito, Ben L. & Larry W. Roberts, *Using Formal Methods to Assist in the Requirements Analysis of the Space Shuttle GPS Change Request,* Langley Research Center, Contractor Report number 4752, August 1996.

33. Easterbrook, S, *Learning from Inconsistency,* in Proceedings 8th International Workshop on Software Specification and Design (IWSSD-8), Paderborn, Germany, NASA-IVV-96-001 WVU-SRL-96-001, 1996.

34. Easterbrook, S, *The Role of Independent V&V in Upstream Software Development Processes,* in Proceedings, 2nd World Conference on Integrated Design and Process Technology (IDPT-96), Austin, TX, NASA-IVV-96-015 WVU-SRL-96-015, 1996.

35. Easterbrook, S, & B. Nuseibeh, *Using ViewPoints for Inconsistency Management,* Software Engineering Journal, 11(1), NASA-IVV-95-002 WVU-SRL-95-002, 1996.

36. Easterbrook, S, & J. Callahan, *Formal Methods for V&V of Partial Specifications: An Experience Report,* in Proceedings, 3rd IEEE International Symposium on Requirements Engineering (RE'97), Annapolis, MD, NASA-IVV-96-007 WVU-SRL-96-007, 1996.

37. Easterbrook, S, & J. Callahan, *Independent Validation of Specifications: A Coordination Headache,* in Proceedings, IEEE 5th Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'96) - Workshop on Requirements Engineering in and for Networked Enterprises, Stanford, CA, pp. 232-237, NASA-IVV-96-013 WVU-SRL-96-013, 1996.

38. Easterbrook, S, & J. Callahan, *SCR as an IV&V Tool,* in Fifth International Software Cost Reduction (SCR) Workshop, Ottawa, Canada, 1996.

39. Easterbrook, S, Rick Convington, John Kelly, Yoko Ampo, & David Hamilton, *Experiences Using Formal Methods for Requirements Modeling,* IEEE Transactions on Software Engineering, NASA-IVV-96-018 WVU-SRL-96-018, 1996.

40. *Feasibility Study: Formal Methods Demonstration Project for Space Applications*, Jet Propulsion Laboratory, Johnson Space Center, and Langley Research Center, a report to NASA Code Q, October 23, 1992.

41. *Formal Methods Specification and Verification Guidebook for Software and Computer Systems, Volume I: Planning and Technology Insertion,* Guidebook (NASA-GB-002-95), 1995.

42. *Formal Methods Specification and Verification Guidebook for Software and Computer Systems, Volume II: A Practitioner's Companion*, NASA, Office of Safety and Mission Assurance, Washington, DC. October 1996.

43. Fotta, Michael, *Reusing Information on Human Functions to Improve Architecture-based System Design,* Software Technology Conference (STC), Salt Lake City, Ut, April 22-28, 1996.

44. *Functional Complexity and Test Efficiency,* Johnson Space Center, July 15, 1995.

45. *Functional Configuration Audit* of SEES (FCATAP:9-1, FCAWKS:9-1), March 1994.

46. *General Evaluation Methodology* of SEES (version 2.3), July 1995.

47. *Guidelines for Measuring S/W Failure Intervals,* Johnson Space Center, September 1994

48. *Guidelines for S/W Measurement as Applied to the HAL/S Language,* Johnson Space Center, March 1994.

49. *Guidelines for S/W Measurement Data Archival,* Johnson Space Center, June 1994.

50. *HAL/S Measurement Tool,* HALMET version 3.0 installation manual, June 1994.

51. *HAL/S Metric Analyzer,* Rev. 3.1, Johnson Space Center, 1994.

52. *Halmet,* version 3.1 software tool, Johnson Space Center, 1994.

53. Hamilton, David, *MIR Docking CR Analysis, Center Initiative Technical Report,* Johnson Space Center, February 4, 1994.

54. Hamilton, David, *Orbit DAP High Level Model Analysis,* Technical Report, Johnson Space Center, February 4, 1994.

55. Hamilton, David, & Rick Covington, *Analysis of FDIR Requirements Using Formal Methods,* Center Initiative Technical Report, Johnson Space Center, July 1,1994.

56. Hamilton, David, Rick Covington, & John Kelly, *Experiences in Applying Formal Methods to the Analysis of Software and System Requirements,* WIFT '95: Workshop on Industrial-Strength Formal Specification Techniques, IEEE-CS, Boca Raton, FL, April 1995, pp. 30-43.

57. *Integration of Metrics with Reliability Predictions,* Johnson Space Center, September 23, 1994.

58. *International Space Station Software IV&V,* Verification, Validation, and Accreditation Colloquium, Sponsored by the Defense Modeling and Simulation Office of the DoD, September 28, 1995.

59. Kelly, John, & Ben DiVito, *Formal Methods Demonstration Project for Space Applications,* 3rd NASA Langley Formal Methods Workshop, May 11, 1995.

60. *Library of Flight Software Components,* Library Augmentation, June 1996.

61. Lutz, R, *Targeting Safety-Related Errors During Software Requirements Analysis,* The Journal of Systems and Software.

62. Lutz, R, & Yoko Ampo, *Experience Report: Using Formal Methods For Requirements Analysis of Critical Spacecraft,* Software Proceedings of the 19th Annual Software Engineering Workshop, Greenbelt, MD, December, 1994.

63. *Mapping S/W Requirements Functionality to Execution Profiles,* Johnson Space Center, June 1994.

64. *Metrics for Risk Assessment, Software Quality, and Process Improvement.* Goddard Space Flight Center, Greenbelt, MD, December 1995, February 1996, March 1996.

65. Montgomery, T, *Design, Implementation, and Verification of the Reliable Multicast Protocol,* West Virginia University, NASA-IVV-95-011 WVU-SRL-95-011, 1994.

66. Montgomery, T, & B. Whetten, *The Reliable Multicast Protocol Application Programming Interface,* NASA/WVU Software IV&V Facility, NASA-IVV-94-007 WVU-SRL-94-007, 1994.

67. Montgomery, T, B. Whetten, & S. Kaplan, *A High Performance Totally Ordered Multicast Protocol,* ICSI, NASA-IVV-94-004 WVU-SRL-94-004, 1994.

68. *A NASA Focus on Software Reuse,* George Mason University, September 23-27, 1996.

69. *NASA, Formal Methods Demonstration Project for Space Applications - Phase I Case Study: Space Shuttle Orbit DAP Jet Select,* Jet Propulsion Laboratory, Document D-11432, December 22, 1993.

70. Neal, R, *The Applicability of Proposed Object-Oriented Software Metrics to Developer Feedback in Time to Impact Development,* in Proceedings of the International Society for Productivity Enhancement: Concurrent Engineering '96, NASA-IVV-96-004 WVU-SRL-96-004, 1996.

71. Neal, R, *Modeling the Object-Oriented Space through Validated Measures,* in IEEE Aerospace Conference '97, NASA-IVV-96-021 WVU-SRL-96-021, 1996.

72. Neal, R, *Software Reuse Metrics and Economics,* in Proceedings of Reuse '96: Reuse as an Integral Part of Software Engineering, 1996.

73. Neal, R, *The Validation by Measurement Theory of Proposed Object-Oriented Software Metrics,* in Proceedings of the 1995 Ernst and Young International Conference of Information Systems Doctoral Consortium, NASA-IVV-96-020 WVU-SRL-96-020, 1996.

74. Neal, R, D. McCaugherty, J. Callahan, & J. Joshi, *Cost Effectiveness of IV&V: A Case Study of Day of Launch I-Load Update (DoLILU) Issue Tracking Reports (DITRs),* Software Engineering Process Group Conference '97: San Jose CA, 1996.

75. Nichols, Dan & Jay Reddy, *Domain Engineering - An Enabling Technology for Software Product Assurance,* ESA 1996 Product Assurance Symposium and Software Product Assurance Workshop, Noordwijk, Nether lands, March 19-21, 1996.

76. Pajerski, R, *Software Process Improvement in NASA's Software Engineering Laboratory,* Software Process Improvement Network Symposium, Hampton, VA, March 22, 1996.

77. Pajerski, R, S. Green, D. Smith, *What's Happening in the Software Engineering Laboratory?, Software Engi neering Workshop, Goddard Space Flight Center, Greenbelt, MD, November 29-30, 1995.*

78. *PCA-RCM Tool,* version 2.0 software tool, Johnson Space Center, 1994.

79. *Physical Configuration Audit* of SEES (PCATAP:10-1, PCAWKS:101), March 1994.

80. *Profile of Software at GSFC,* June 1994.

81. *Profile of Software Within Mission Operations and Data Systems Directorate at the Goddard Space Flight Center,* December 1992.

82. *Profile of Software within NASA,* March 1995.

83. *Report on an Experiment in Including Metrics in a Software Reliability Model,* Johnson Space Center, 1994.

84. *Requirement Metrics for Risk Identification,* Software Engineering Laboratory Workshop, Goddard Space Flight Center, December 1996.

85. *Requirements Review & Assessment* of SEES (RRATAP:1-1, RRAWKS:1-1), July 1993.

86. *Requirements Trace/Completeness Matrix* of SEES (RTMTAP:2-1, RTMWKS:2-1), May 1994.

87. *Results of Reliability Model Application,* Johnson Space Center, March 1-15, 1995.

88. Roberts, Larry W. & Mike Beims, *Using Formal Methods to Assist in the Requirements Analysis of the Space Shuttle HAC Change Request,* (CR 90960E), NASA Johnson Space Center, 1996.

89. *S/W Maintainability & Reliability Measurement Criteria,* Johnson Space Center, August 1994.

90. *SEES Executive Summary,* August 1994.

91. *Software Management Guidebook,* June 1996.

92. *Software Measurement for Practitioners Course,* September 1996.

93. *Software Measurement Guidebook,* August 1995.

94. *Software Metrics for Risk Assessment,* 47th International Astronautical Congress & Exhibition, 29th Safety and Rescue Symposium, October 1996.

95. *A Software Metrics Model for Projecting Risks and Assessing Quality,* European Space Agency Software Assurance Symposium, Netherlands, March 1996.

96. *Software Process Improvement Guidebook,* January 1996.

97. *Software Quality Metrics for Object-Oriented Environments,* Unisys Technology Conference, VA, August 1996.

98. *Software Quality Metrics for Object-Oriented System Environments,* Goddard Space Flight Center, Software Assurance Technology Center Technical Report, SATC-TR-95-1001, June 1995.

99. *A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality*, Reprinted in Society for Software Quality Journal, April 1996.

100. *Software Re-Engineering,* Goddard Space Flight Center Software Assurance Technology Center Technical Report, SATC-TR-96-1001, October 1996.

101. *Software Reliability Assessment Toolset,* Johnson Space Center, September 30, 1995.

102. *Software Reuse Metrics and Economics,* 3rd annual meeting of the Software Reuse Metrics Working Group, Morgantown, WV, July 1996.

103. *The Statistical Testing of the Space Shuttle Primary Avionics Software System,* Johnson Space Center, September 1, 1994, February 22, 1995.

104. *Summarize the Progress in Exploring the Feasibility of Modifying the Way in Which the Schneidewind Reliability Model Parameters are Estimated,* quarterly report, April 1, 1994 - June 30, 1994.

105. *Test Description Assessment* of SEES (TDATA:6-1, TDAWK:6-1), January 1994.

106. *Test Witness & Assessment* of SEES (TWATAP:7-1, TWAWKS:7-1), May 1994.

107. Tran, Lan, *The Evaluation of V & V Tools and Concepts for Autonomous Spacecraft Operations,* Jet Propulsion Laboratory/Executive Council, September 16, 1996.

108. *Trends in Complexity in Software Development,* Johnson Space Center, August, 1994.

109. Wen, W, & J. Callahan, *Neuralware Engineering: Develop Verifiable ANN Systems,* in Proceedings of IEEE Joint Symposia on Intelligence and Systems, Washington DC, November 4-5, 1996.

110. Wen, W, & J. Callahan, *Verification and Validation of Knowledge Based Systems with ANN Components,* in Proceedings of AAAI'96 Workshop in Verification and Validation of KBS and its components, Portland, OR, August 3-8, 1996.

111. Wen, W, J. Callahan, & M. Napolitano, *Design Verifiable ANN-based Flight Controller,* in Proceedings of ICTAI '96 Workshop on Artificial Intelligence for Aeronautics and Space, Toulous, France, Nov. 13-16, 1996.

112. Whetten, B, T. Montgomery, & J. Callahan, *The Reliable Multicast Protocol Specification: Protocol Operation,* NASA/WVU Software IV&V Facility, NASA-IVV-95-004 WVU-SRL-95-004, 1995.

113. Whetten, B, T. Montgomery, & J. Callahan, *The Reliable Multicast Protocol Specification: Protocol Packet Formats,* NASA/WVU Software IV&V Facility, NASA-IVV-95-003 WVU-SRL-95-003, 1995.

114. Woodhouse, Robert M, & R. Lutz, *Contributions of SFMEA to Requirements Analysis,* in Proceedings of the 2nd IEEE Conference on Requirements Engineering, 1996.

115. Woodhouse, Robert M, *Requirements Analysis Using Forward and Backward Search,* Annals of Software Engineering Special Volume on Requirements Engineering, 1996.

116. *Workshop Version* of SEES (TRAPAP:2-1, TRAWKS: 2-1), June 1993.

117. Zelkowitz, M, *Software Engineering Technology Infusion Within NASA,* IEEE Transactions on Engineering Management, vol. 43, no. 3, August 1996.

118. Zelkowitz, M, & R. Tesoriero, *Process Enactment within an Environment*, Software Engineering Workshop, Goddard Space Flight Center, Greenbelt, MD, November 29-30, 1995*.*